Improving Knowledge Management in Software Reuse Process[1]

Timo Kucza, Minna Nättinen, Päivi Parviainen

VTT Electronics, PL 1100, 90571 Oulu, Finland
{Timo.Kucza, Minna.Nattinen, Paivi.Parviainen}@vtt.fi

Abstract. We demonstrate the need for reuse in today's software development organisations in the light of widely accepted literature. Based on this need, we introduce the reuse processes and identify the knowledge dependent processes involved. We provide an introduction to the KM process model developed at VTT and describe the process of continuous improvement integrated to it. We establish a link between reuse and KM and then develop a theory of KM process improvement. Finally we give an outlook on a coming customer project in which we are going to evaluate this theory in an industrial case.

1 Introduction

Knowledge is essential in everyday work. Everyone learns by experience and this knowledge is later reused in similar kind of tasks again by adapting this knowledge to a new situation. The general purpose of Knowledge Management (KM) is to make this kind of knowledge usable for more than one individual, e.g. for an organisation as a whole, i.e. to share it. Techniques and practices similar or even identical to KM have been used for a long time, although they have neither been called by this name nor necessarily recognised as knowledge management until a few years ago [1]. Due to the steadily increasing speed at which new techniques are being developed along with the need of integrating knowledge into processes and products, knowledge is, at present, more and more widely considered as the most important asset of organisation [2]. Various industrial projects have proven that KM has potential to remarkably improve processes and products [3].

Contrary to prior assumptions, KM concerns all organisations [4]. The field of software development is especially affected, because of the continuing struggle to stay abreast of new technologies, to deal with frequently changing customer requirements and more and more complex software products, and to cope with the competition in highly demanding markets. Some issues can be addressed by distributing organisations geographically or even organisationally, but with these new management techniques also new problems arise. Many software development organisations have begun to understand that if they are to succeed in the future, they need to manage and use more effective, productive, and innovative knowledge on individual, team and organisational levels [1, 5]. Software Process Improvement (SPI) is the most general method used for meeting these changing challenges in software development, and a large amount of profitable research and practical work has been done in this area [6, 7]. A promising, but not yet fully exploited method of improving the process of software development can be seen in the implementation of reuse [8].

The purpose of this paper is to propose an approach to applying KM to software reuse process improvement. While software itself is knowledge intensive [9], reuse has been chosen as the application domain here, because it amplifies the need for proper mediation of knowledge not only between people but also over time. The concepts KM and reuse are similar to a great extent - while reuse commonly stands for the reuse of software components, KM denotes the reuse of knowledge. The difference lies in the scope and approach of the concepts. Whereas KM has a wide scope, as almost everything can be regarded as being based on knowledge, reuse is more focused on components appearing in software development. While KM puts an emphasis on tacit knowledge, reuse focuses on explicit knowledge as embedded in reusable components. We intend to show that the combination of the two concepts can bring the underlying idea of using already existing assets to its full potential.

The results presented here are going to be used in a case project, which will provide a verification of the usability of this concept in industrial practice. The underlying work is conducted as a co-operation of two projects at the Technical Research Centre of Finland (VTT): a strategic research project focusing on KM and a SPI project for a customer.

---

[1] Published in: Bomarius, Frank & Komi-Sirviö, Seija (Eds.): Product Focused Software Process Improvement: 3rd International Conference, PROFES 2001, Proceedings, pp. 141-152. Goos, Gerhard; Hartmanis, Juris & van Leuwen, Jan (Series Eds.): Lecture Notes in Computer Science, Vol. 2188. Springer: Berlin, Heidelberg, New York, 2001.

We provide background information about the underlying concepts and related terms of KM and reuse in the following and introduce the KM management processes in chapters 2. The process model for KM that we are currently developing will contain an improvement cycle, which is considered important for the success of any related projects. This improvement process is described in chapter 3. In chapter 4, we bring together the methodologies of KM and reuse. Finally, the conclusions of this work are presented in chapter 5.

1.1 Reuse

Reuse can be defined as further use or repeated use of an artefact (or asset). The idea of reuse in software development was first introduced by M. D. McIlroy in 1969 [10] when he proposed a catalogue of software components from which software could be assembled. Since then the software industry has grown exponentially and during this time there have been several reports describing the benefits achieved through reuse, e.g., [11, 12]. However, it has been widely accepted that software reuse offers yet unrealised potential for enhancing software development productivity, and improving the quality and reliability of the end products [8]. From a theoretical point of view, reuse as a development concept offers solutions to many problems in industry. In practice, however, many companies have gained less than expected, and the promises of software reuse remain for the most part unfulfilled [13]. Implementing software reuse has a number of pitfalls, such as making the focus too small, not utilising procedures adopted for reuse, or missing knowledge on how to use the existing possibilities [8]. As a company-wide venture it also carries certain risks. Still, the potential benefits of software reuse have made it one of the most interesting practices that are yet to be fully utilised in software development.
Reuse is commonly divided into two main activities: for reuse and with reuse. For reuse means development of reusable assets and with reuse means building new products by reusing these assets. Fig. 1 illustrates reuse activities and their relation to one another.

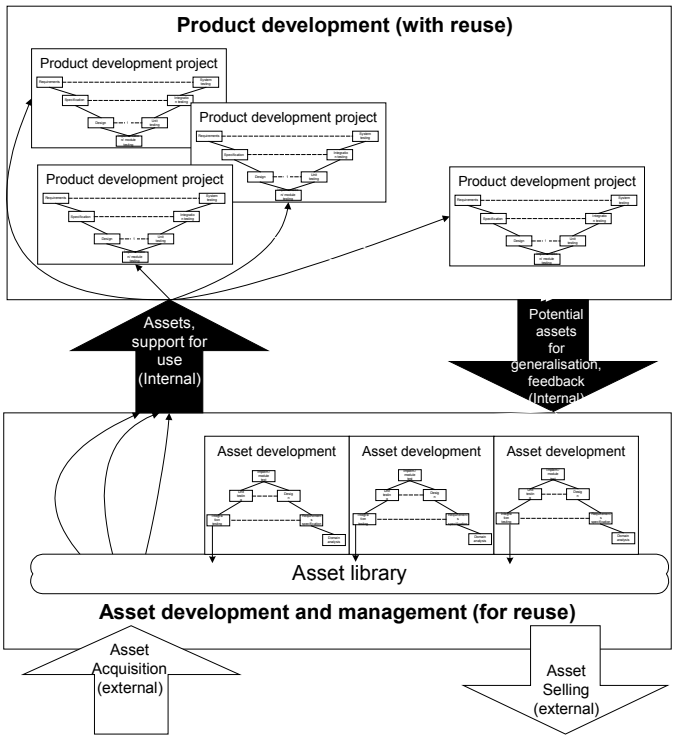

Fig. 1. Reuse activities

The general steps to follow in the development for reuse in our experience are (adapted from [14, 15]):

1. Collect requirements from potential reusers.
2. Analyse requirements and domain.
3. Select and prioritise requirements.

4. Develop and document the solution for selected requirements.
5. Evaluate the asset (testing).
6. Store asset with documentation.
7. Support reusers.

In with reuse, a new product is created by taking applicable assets from the asset base, tailoring them as necessary through pre-planned variation mechanisms, such as parameterisation, adding any new assets that may be necessary, and assembling the collection. The general steps of the with reuse project in our experience are (adapted from [14, 15]):

1. Find candidate assets for reuse from asset base.
2. Evaluate the candidate assets with respect to requirements.
3. Select assets to be reused.
4. Adapt/modify the assets, if needed.
5. Integrate assets into product.
6. Test.
7. Give feedback to for reuse development.

Reuse also affects supporting processes, that is, the practices needed for supporting the engineering activities. These practices include, e.g.,

- Documentation; assets must be well documented; the greater the asset, the greater the role of documentation. Reusers should give feedback on assets as this information is valuable for the developers and maintainers of assets.
- Quality assurance; quality problems must be identified and dealt with both when developing new assets and when maintaining existing assets.
- Project management; the major difference between managing a project involving reuse and a normal one is the increased number of external stakeholders in the former case, leading to more negotiations and complex trade-offs, as approval is required from all parties (i.e., reusers, developers and management) [15].

1.2 Knowledge and Knowledge Management

There are a number of different definition types for knowledge in literature (see [16] for a summary). The viewpoint common to all of these definitions is that knowledge is something more than just data and information. While data can be defined as a set of facts, and information as a meaningful message, knowledge denotes a combination of human context, experience, values, and expert insights [1]. What this brief definition immediately makes clear is that knowledge is by no means straightforward or simple. Knowledge appears at individual, team, and organisational level, and it exists among people who are affected by the organisation culture, i.e. behavioural norms and practices [17]. A distinction is commonly made between internal tacit knowledge and external explicit knowledge, e.g. as codified into documents, processes etc. [5]. Knowledge is what enables organisations to operate. Or, in other words "…what an organisation and its employees know is at the heart of how the organisations function" [1]. KM, as we use it here, is the activity of managing the creation, sharing and utilisation of knowledge. As knowledge is something complex and mostly internal to human minds, managing the environment is of great importance here. Thus KM deals with building environmental conditions, such as culture, before implementing any concrete sharing processes. The differentiation between knowledge and information is a complex issue [18]; we simply assume that KM covers Information Management (IM), thus also addressing the reversed knowledge hierarchy as introduced by [19].

2 Knowledge Management Process

Several SPI models exist for the field of software engineering (see [20] for an overview) such as the Pr^{2}imer model [21, 22] developed at VTT. However, when approaching the topic KM in general, we realised that no detailed process model, such as the V-Model for software development [23], was available for KM. We concluded to approach KM by starting work on such a process model. In the case described in this paper the developed model is utilised in an industrial environment, which provides verification of its usability.

In this section, a brief introduction is given to the KM Process Model developed at VTT. The processes, 39 of them altogether, are determined in detail. In this introduction, only the main processes of KM are described. Fig. 2 shows the main processes and their interaction between each other.

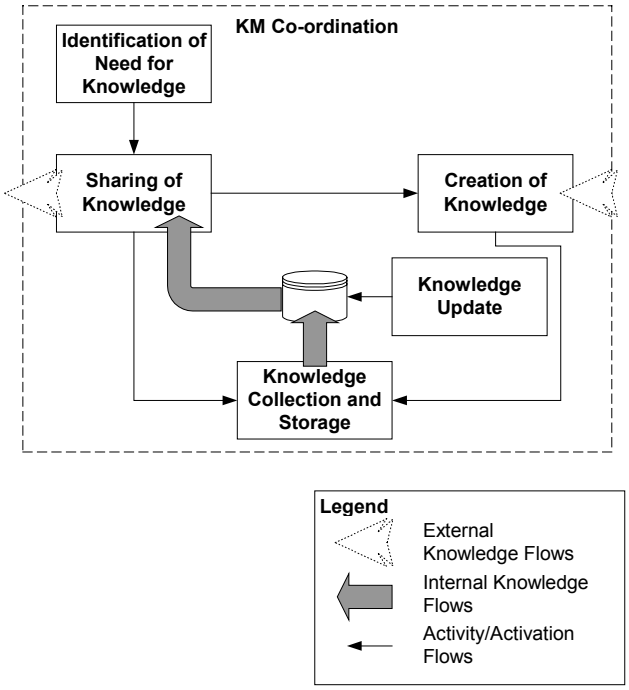

Fig. 2. Main KM Processes

In the following, a short description of each of the main processes is given:

- KM Co-ordination: The current state is analysed and a target state is defined as goals and the progression planned in form of processes on basis of the scope of the KM project. The sub-processes constitute an improvement cycle for a continuous revision of the processes.
- Identification of Need for Knowledge: Before existing knowledge can be shared, the need for knowledge has to be identified and the requirements for the desired knowledge have to be determined.
- Sharing of Knowledge: Whenever knowledge as desired is already present, it should be shared. In addition, knowledge can be shared to the outside world (Knowledge Brokering). Whichever the case, knowledge always has to be set into a proper context. If knowledge is applied to a new context, the result may be new knowledge that needs to be considered for collections and storage.
- Creation of Knowledge: If the needed knowledge does not yet exist, it needs to be created. Knowledge creation denotes a combination of knowledge either from inside the organisation, or from the outside (Knowledge Acquisition), or both. External knowledge is connected to Knowledge Creation because of the need to adopt it to the organisational context.
- Knowledge Collection and Storage: New knowledge is not only created by demand, but also through inventions and improvements throughout the work. Whenever new knowledge is created, it needs to be identified. The created knowledge can then be evaluated so as to determine whether it will be of further use. If so, it needs to be stored to make sharing possible.
- Knowledge Update: Knowledge is context-specific and thus has to change with changing environment. This requires the knowledge stored to be kept up to date. Thus, the changes affecting stored knowledge need to be identified so that they can be analysed towards their impact on the stored knowledge. If needed, appropriate changes need to be undertaken, while outdated knowledge also needs to be discarded.

These knowledge processes are always performed in some way already, which usually is an informal, unconscious and non-technical one. The general goal of a KM project can be formulated as making these processes known to the persons involved, improving them where shortcomings can be identified, and assisting them with technical means where possible and reasonable. Basically, however, KM mainly depends on humans. Therefore, cultural and behavioural aspects are the most important ones, whereas technical means play a secondary role [1]. Nevertheless, technological assistance is regarded as inevitable, once the enabling environment has been established. Especially for address distribution issues, computer-mediated communication and transfer of knowledge are of great importance.

3 KM Process Improvement Model

In this chapter, the KM process improvement cycle is introduced. The main phases of the model are based on Pr^{2}imer process improvement model: Analysis, Definition, Planning, and Effecting. In our process model, this cycle is integrated to the main process KM-Co-ordination. The resulting cycle is shown in Fig. 3.

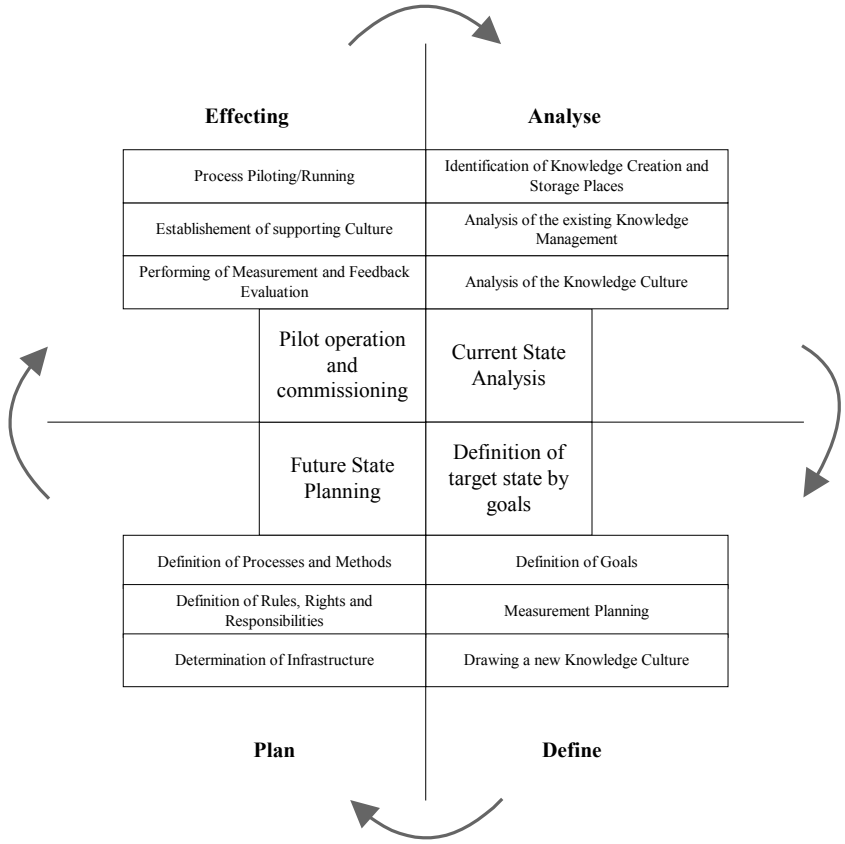

Fig. 3. KM Improvement Cycle

This improvement cycle addresses several issues [1]:

- the need to incrementally approach changes,
- the problem of fast changing technology and working practices [2] and the resulting danger of out-of-date knowledge,
- the poor predictability of costs of KM activities which impedes management decisions, and
- the difficulty of influencing cultural and behavioural practices.

Starting KM with piloting, improving the processes until they work properly, and then widening the area of influence enable a gradually ongoing development instead of a radical change. This allows people to get used to new working practices, and a new cultural environment can be build step-by-step. Also the costs for KM activities can be kept at an acceptable level until success can be seen (and measured) and thus larger investments become feasible. From the start on, the monitoring and measurement features enable the KM system to keep up with any changes and to avoid outdated knowledge that would endanger KM as a whole [1].

4 KM Improvement in the Software Reuse Process

Based on the findings described above, our approach to an industrial case is introduced in the following. Since the case is in its early phases, this paper will focus on the analysis phase and only an outlook to the other phases of the KM improvement cycle will be given.
In order to improve reuse processes by means of KM process improvement, the reuse and KM process areas that need improvement should be identified first. Thus, the improvement process starts with a current state analysis, in which the reuse processes are assessed, with a special focus on knowledge.

4.1 High Level Analysis of Current State (Analyse)

First, the existing reuse process will be analysed at high level in order to identify knowledge intensive areas in the process for KM improvement. This analysis is done via interviews and workshops. The general steps of the reuse process presented in chapter 1 are addressed. Also, the potential risks in the reuse process are analysed to find out what needs to be addressed in the knowledge management activities.
The results of the general level analysis are evaluated in a workshop together with the reuse process owner, the quality manager and managers from both product and asset development projects. The evaluation results are then used for setting the focus for the detailed analysis of the affected reuse processes, emphasising the knowledge viewpoint. Example focus areas include:

- The knowledge sharing between product development (with reuse) and asset development and management (for reuse). This is especially important if different people in different locations perform these tasks. For example, knowledge concerning future directions for products is needed when evaluating the reuse potential of an asset candidate. When new products are assembled from the defined reuse assets (for reuse), knowledge of the design rationale of these assets will be needed. Also, the so-called NIH factor (Not-Invented-Here) may prevent successful reuse.
- General communication about reuse strategies etc.
- Knowledge sharing between with reuse projects, e.g., how certain assets worked, where they can be used etc.
- Searching for appropriate assets from the asset library.

Improvement actions are performed using incremental approach in order to get some visible benefits early and thus, to keep the people involved motivated and to provide feedback for the management. Number of increments and effort and schedule for the increments depends on increment's focus and are planned separately for each increment. The focus area for the first cycle is agreed during this high level current state analysis. Each increment consists of the following tasks:

- Detailed analysis of current state (strengths and needs for improvements).
- High level description of target state.
- Drawing up an improvement plan for selected findings including improvement actions, responsibilities, schedule, improvement actions follow-up, metrics, etc.
- Implementation and evaluation of improvement actions.
- Planning the next increment (focus, schedule, and effort).

4.2 Detailed Analysis of Current State (Analyse)

Detailed analysis will be performed via interviews and study of the processes involved. Firstly, the organisational units involved have to be identified and the interviewees determined. In the example focus area of the interface between for and with reuse the interviewed persons will be those project managers and designers of product projects who are using the reusable SW assets in product development.
The interview questions are defined based on the analysis focus and KM process definitions. Each of the KM process areas (KM Co-ordination, Identification of Need for Knowledge, Sharing of Knowledge, Knowledge Collection and Storage, and Knowledge Update) is taken into account when drawing up the interview questions. In the detailed analysis the following questions are addressed, for example:

- What knowledge needs to be managed to address the problem?
- Where is this knowledge located?
- At which points of the reuse processes is this knowledge needed?
- What is hindering the knowledge from being used where it is needed?

The importance of social and cultural aspects for successful KM requires an emphasis on corresponding issues in the organisation, i.e. the environment has to be analysed and possibly influenced towards a supporting culture.
In the example focus area of the interface between for and with reuse the questions to be addressed in interviews include, e.g., the following:

- Are reusable assets searched for during product development? At what point of the development process is this done? How is this search performed? What information is needed in performing the search? Can suitable assets be found with reasonable effort? What problems occur?
- How much adaptation is required for the current needs? Are the assets reusable with reasonable effort? Are the costs predictable for the required adaptation? What information is needed in order to be able to make the needed adaptations? How is adaptation performed? What problems occur?

Here the knowledge sharing process would be of main interest in the interviews. For example, the problem that reuse is not performed, can thus be refined to its roots, e.g., searching is not performed (culture), assets can not be found with reasonable effort (insufficient descriptions or search capabilities), or adaptation is too complicated (improper codification).
In addition to interviews, an analysis of documentation needs to be performed. This enables gaining an understanding of how processes are planned (quality manual) and how processes are supposed to be performed (project reports), thus bringing an additional viewpoint to the interview results.
As a result the strengths and weaknesses in Knowledge Management are identified for the reuse process in the selected focus area.

4.3 Definition of Target State (Define)

In order to be able to plan the improvement actions an ideal or better situation needs to be drawn up for the selected focus area. Therefore, the goals for improvement are defined based on the improvement ideas found during the current state analysis. This will be carried out in co-operation with the reuse project members. The processes designed are regarded as initial and need to be checked against improvements once they are used.
Measurements are then defined to support monitoring the success of improvements by refining the goals to questions and these to metrics according to the GQM-Paradigm [24].

4.4 Improvement Planning (Plan)

The identified improvements are defined as new processes and responsibilities in the improvement plan. In the example focus area of the interface between for and with reuse improvement activities could be the following:

- Introduction of meetings between for reuse and with reuse departments at certain phases of each development project,
- Introduction of general communication channels between the two departments in form of regular meetings, videoconferencing, newsgroups, and newsletters.
- Introduction of a knowledge base designed to address the identified problems to provide easier access to the assets.

The outcome of this stage is highly dependent on the prior stages and of the focus area.

4.5 Implementing and Evaluating the Improvement Actions (Effecting)

The improvements are carried out as a pilot operation according to the improvement plan. The introduction of new techniques and methods is performed step by step, which makes it easier for the people involved in the processes to accept and adapt to the changes. Through this stage-wise approach, acceptance can be easily reached among the people affected, which is elementary for the success of the project. Metrics are used regularly in the processes to enable active tracking of performance and identification of possible problems.

5 Conclusions and Further Work

We have introduced a KM process model and its improvement cycle at a high-level. We have shown our ideas on how our KM model can be applied to reuse processes. Emphasising the analysis phase of the KM improvement cycle we have introduced how we are planning to perform the first KM-based reuse process improvement sequence in a case organisation.
The scenario depicted here, although affecting the basic reuse processes, does not bring any actual changes to them. What we are suggesting in this paper is not a new reuse process model, but a reuse process enhanced by KM processes, which enables addressing problems occurring in reuse. The first improvement cycle we have introduced is merely intended as the first piloting of these means. Any shortcomings identified during the pilot operation can be addressed in further improvement cycles. In addition, an extended focus can be applied to widen the area of impact. Each further improvement step concerning content or scope will take a cycle of its own in the KM improvement process. This ensures that the most important aspects are taken into account and addressed accordingly.
The methodology of KM-based reuse process improvement allows piloting and improving the reuse process, and once changes are working spreading them for wider use. The general aim being a process of continuous improvement, we are planning to perform the first one or two cycles together with the customer, assuming that the following cycles will be performed without us. However, we will be monitoring this activity so as to be able to verify the applicability of the developed KM process model and to address any shortcomings on the methodology identified during this work.

References

[1] Davenport, T.H. and Prusak, L. Working Knowledge - How Organizations Manage What They Know. Boston, Massachusetts: Harvard Business School Press; 1998. ISBN 0-87584-655-6.

[2] Carneiro, A. How does knowledge management influence innovation and competitiveness? in: *Journal of Knowledge Management*; Vol. **4** (2). pp. 87-98.

[3] O'Dell, C. and Grayson, C.J., Jr. If Only We Knew What We Know: The Transfer of Internal Knowledge and Best Practice. New York: The Free Press; 1998. 238 pp. ISBN 0-684-84474-5.

[4] Marler, K. Rapid Emerging Knowledge Deployment in: *Crosstalk: The Journal of Defense Software Engineering*; Vol. **12** (11). pp. 14-16.

[5] Nonaka, I. and Takeuchi, H. The Knowledge-Creating Company. Oxford, New York: Oxford University Press; 1995. 284 pp. ISBN 0-19-509269-4.

[6] El Emam, K.; Drouin, J.-N. and Melo, W. SPICE: The Theory and Practice of Software Process Improvement and Capability Determination. Los Alamitos: IEEE Computer Society; 1998. 486 pp. ISBN 0-8186-7798-8.

[7] Fitzgerald, B. and O'Kane, T. A longitudinal study of critical success factors for software process improvement in: *IEEE Software*; Vol. **16** (3). pp. 37-45.

[8] Jacobson, I.; Griss, M. and Jonsson, P. Software Reuse: Architecture and Organization for Business Success. New York: ACM Press; 1997. 497 pp. ISBN 0-201-92476-5.

[9] Hilburn, T.B.; Hirmanpour, I.; Khajenoori, S.; Turner, R. and Qasem, A. A Software Engineering Body of Knowledge. Carnegie Mellon University (1999). Online: http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr004.pdf.

[10] McIlroy, M.D. "Mass produced" Software Components in: P. Naur and B. Randell, Editors: *Proceedings of the 1968 NATO Conference on Software Engineering*. Brussels: NATO Scientific Affairs Division; 1969. pp. 138-155.

[11] Lanergan, R.G. and Poynton, B.A. Reusable Code - The Application Development Technique of the Future in: *Proceedings of the SHARE/GUIDE/IBM Applications Development Symposium*. Monterey, California: IBM; 1979. pp. 127-136.

[12] Tajima, D. and Matsubara, T. Inside the Japanese Software Industry in: *Computer*; Vol. **17** (3). pp. 34-41,43.

[13] Mili, A.; Yacoub, S.; Addy, E. and Mili, H. Toward an Engineering Discipline of Software Reuse in: *IEEE Software*; Vol. **16** (5). pp. 22-31.

[14] Lim, W.C. Managing Software Reuse, A Comprehensive Guide to Strategically Reengineering the Organization for Reusable ComponentsPrentice Hall; 1998.

[15] Karlsson, E.-A. Software Reuse: A Holistic Approach. Chichester: John Wiley & Sons; 1995. 510 pp. ISBN 0-471-95489-6 / 0-471-95819-0.

[16] Spiegler, I. Knowledge Management: A New Idea or a Recycled Concept? in: *Communications of the Association for Information Systems*; Vol. **3** (

[17] De Long, D. Building the Knowledge-Based Organization: How Culture Drives Knowledge Behaviors. Working Paper, Cap Gemini Ernst & Young Center for Business Innovation, 1997. Online: http://www.businessinnovation.ey.com/mko/pdf/wculture.pdf (accessed: 2001-02-19).

[18] Leonard, D. and Sensiper, S. The Role of Tacit Knowledge in Group Innovation in: *California Management Review*; Vol. **40** (3). pp. 112-132.

[19] Tuomi, I. Data is more than knowledge: Implications of the reversed knowledge hierarchy for knowledge management and organizational memory in: *Proceedings of the 1999 32nd Annual Hawaii International Conference on System Sciences*. Los Alamitos, California: IEEE Computer Society; 1999. pp. 45-.

[20] Rada, R. and Craparo, J. Standardizing Software Projects in: *Communications of the Association for Computing (ACM)*; Vol. **43** (12). pp. 21-25.

[21] Pr²imer. Webpage, Technical Research Centre of Finland (VTT), 1999. Online: http://www.vtt.fi/ele/research/soh/products/primer_etusivu.htm (accessed: 2001-04-18).

[22] Karjalainen, J.; Mäkäräinen, M.; Komi-Sirviö, S. and Seppänen, V. Practical process improvement for embedded real-time software in: *Quality Engineering*; Vol. **8** (4).

[23] The V-Model. Webpage, German National Research Centre for Information Technology (GMD), 1996. Online: http://www.scope.gmd.de/vmodel/en/ (accessed: 2001-03-19).

[24] van Solingen, R. and Berghout, E. The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development. London: McGraw-Hill; 1999. 199 pp. ISBN 0-07-709553-7.