Process Models, Integrated CASE-Tool Environments and Project Management - Birds of a feather or enemies for life?

Gerhard Chroust, Susanne Hofer
Kepler University Linz

Abstract

The need to improve software development, especially with respect to quality and productivity, has been approached from many different angles, the most prominent ones being Process Models, Integrated CASE-Tool Environments, Repository, Project Management Tools and Process Engines. Environments containing some or all of these components are on the market. Still fully integrated environments, where all these components have equal influence seem to be lacking.

In this paper we will explore the interfaces between the components of a software engineering environment. We speculate why process-centered software engineering environments seem to lack market acceptance. The paper is based on the authors' experience with ADPS, ADW, KEY, PROSIMULA, MAESTRO/II and winGBS.

Basic questions to be posed and up to now only partially answered are:

- What are the problems when trying to integrate a process model with an integrated CASE-Tool environment?
- What is the reason that Repository and Process Control Store do not harmonize with one another?
- Can the control by a Process Model be unobtrusive?
- Why do Project Management and Process Models speak a different language?

1 State-of-the-Art

Approaches to managing software and their support by computers (Software Engineering Environments [Hausen-81a, Hausen-85, Chroust-92a]) have had several different starting points and objectives. When discussing such environments essentially five basic components can be distinguished (Fig. 1):

Process Models describing the process without too much consideration of available tools. Typical representatives are the German V-Model [Broehl-92, Broehl-93], the late ADPS [Chroust-88g, Chroust-89d], SSAMD [Downs-88, Hares-90], MAESTROII [Merbeth-89, Yourdon-90], Merise [Deltl-89], HERMES [BAInf-95] etc. The new draft standard ISO 15504 (formally SPICE [Dorling-93, Rout-95, Simon-96]) has also that flavour.

Process Engine It is a natural idea from manufacturing to enforce/enact a defined process via an appropriate interpreter. It provides for the user the access to various components of the environment, interprets the Process Model (if included), gives access to the tools etc. Typical examples are ADPS/M [Chroust-89d], Process Weaver, winGBS and a few others [Hausen-85].

Repository A major idea from Information Engineering [Martin-89a] is to keep all data relevant to the system development process (and even more relevant to the operation of the company) in a common Repository [Kay-90]. All intermediate and final results and their relationships are stored there [Habermann-93, Sagawa-90]. The Repository was intended to be the heart and basis of AD-CYCLE [Corzilius-92, Mercurio-90].

CASE-Tools Tools are the workhorse of software engineering. Again James Martin [Martin-89a] provides eloquent arguments for the automation of the system development process.

Project Management Tools The technical development process is only part of a larger process. A major concern is management of resources (time, money, people), and providing schedules, projections, estimates etc. Essentially they rely on a work breakdown structure which very often has little resemblance to the activities in the technical development area [CTN-95, Ebner-95, Hazeyama-92, Hofer-95].

Given the above 5 major components we see various combinations of them, largely depending on the starting point of the respective vendor (as already has been pointed out by [Hausen-82]. Currently we see two major manifestations:

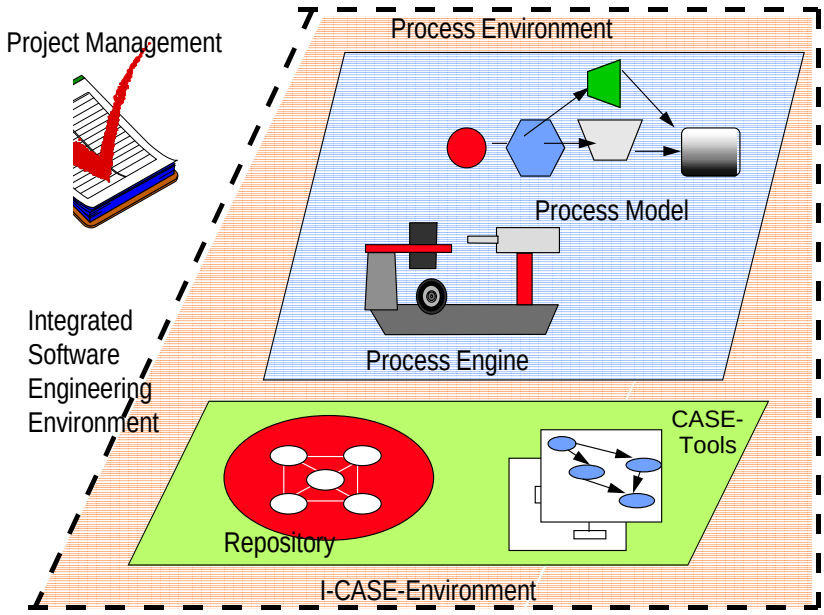

Figure 1: Starting Points for IPSEs

Integrated CASE-Tool Environments These provide an integrated and comprehensive CASE-Tool Environment set usually covering all phases of the software life cycle together with an integrating repository. Many of them are essentially James Martin's children,

implementing the idea of Information Engineering. They constitute a set of tools which propagate a common philosophy, have a common 'look and feel' and are usually integrated via a repository [Kay-90].

- KEY, previously ADW and even before (with a change of environment) IEW [Kay-90, Privateer-90]
- Composer, previously IEF
- SDW
- Silverrun
- etc.

Process-centered Software Engineering Environments Usually they are mainly concerned with enacting the process model (also a challenge for the academic world, cf. [Dowson-91a, Dowson-86b, Dowson-94, Chroust-94b]). They usually provide a well-defined process model and an interpreter. The tool support seem to be lagging behind and the project management modelling being rather poorly modelled (section 3 . Examples are MAESTROII [Merbeth-89, Yourdon-90], ADPS [Chroust-89d], and also Process Weaver.

With AD-Cycle [Corzilius-92] IBM tried to combine an integrated set of CASE-Tools with a software process model, a support environment and a Repository. The basis was the process engine ADPS/P and a process model, called ADPS/M [Bandat-90, Chroust-89d, Chroust-90b, Chroust-91e, Chroust-94d, Corzilius-92].

Despite all the perceived differences a successful software project needs all components.
Basic questions to be posed and up to now only partially answered are:

- Why are these aspects treated separately?
- Why is there so little common ground of understanding?
- Why is there so little data interchange?
- What are the problems when trying to integrate a process model with an integrated CASE-Tool Environment?
- Can the control by a Process Model be unobtrusive? [Chroust-93h]
- What is the reason that Repository and Process Control Store do not harmonize with one another?
- Why do Project Management and Process Models speak a different language? [Chroust-92g, Hofer-95]

2 Process Models and CASE-Tool Environments

Basically a process model describes the activities to be performed during a software development project. For many of these activities specific tools exist. Thus the tool activation can be largely done based on the information in the process model and the current state of the development project (contained in the status of the artefacts in the repository [Chroust-91a, Chroust-91e]).

- A problem arises from exchanging relevant process information and often from an over-emphasis of the importance of the process model. The process model is felt to be a hindrance and not a help by many experienced software engineers. The cry for an 'unobtrusive' process control has been heard, but not answered yet.
- Another problem [Garra-95] is caused by the flexibility of current integrated CASE-Tools allowing the user to switch subtools (diagrammers in KEY terminology) without informing the process engine. Thus central control might be lost [Privateer-90], cf. Fig. 2 .

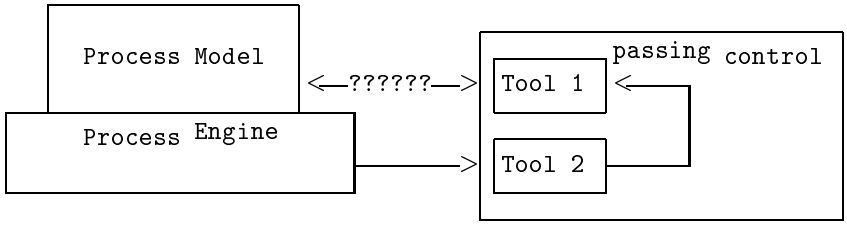

Figure 2: Evasion of control from the process engine

.

- A third problem [Garra-95] is that these integrated CASE-Tool Environments have a Repository of their own, usually proprietary, and it is almost impossible to extract the appropriate information on result changes from there.

3 The Process/Project interface

Essentially Process Models and Project management should be seen as 2 sides of a coin [Berkeley-90]. While process models define the *logical* constraints and the technological strategy of development, project management limits the available options to those feasible under the given resource contraints.
When considering a project managers view of the development process one realizes that the granularity of a project manager's planning is much coarser than the granularity usually provided by process models. Thus a mapping between the existing (or planned) activities and the planning units of the project manager (the so-called 'tasks') has to be established. Part of the problem is that the assignment of activities to tasks (apart from being many-to-one) is completely up to the project planner and is influenced by factors not known to the process engine.
Fig. 3 demonstrates some of the problems of interfacing process management with project management.

- Project management considers work packages ('tasks') which are usually of collection of different activities, even of different type.
- Project management has to plan the instances of work, despite the fact that at planning time the *number* of instances might not be known and no instantiation might have taken place yet.
- Project management has to cope with rework, a typical feature in software engineering.

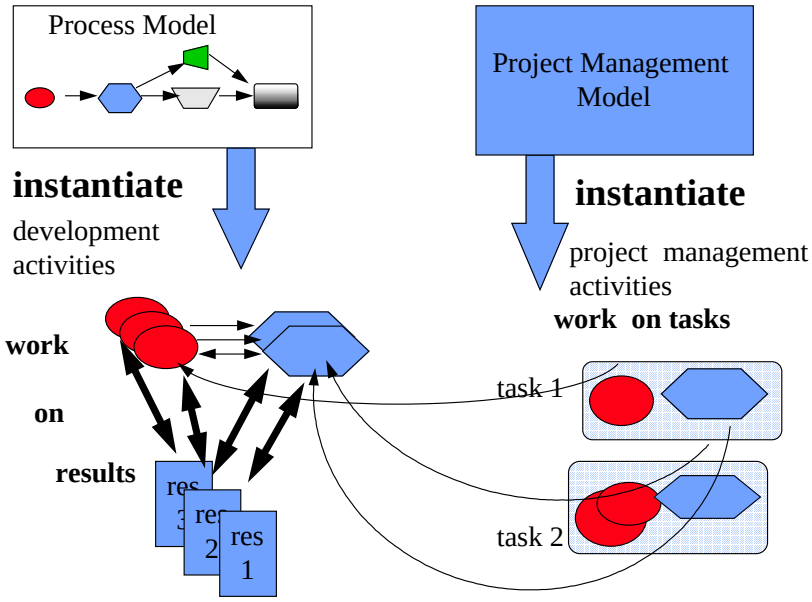

Figure 3: Relation between Process and Project

4 Where does the trend go to?

Watching the development in the software arena we can distinguish several trends which will influence the view and also the economic success of the future software engineering environments.

Standardisation of Software-Processes ISO 12207 defines the Software-Processes on a very high-level [ISO-12207]. In SPICE (now draft for ISO 15504) the processes are still more detailed. The description, however is such, that a direct enactment by a process interpreter is impossible.

Capability levels and certification Software houses will be required to demonstrate their (cap)ability to produced the promised piece of software. This involves (already for ISO9000 certification [Bailetti-95, ISO-94, Kehoe-96, Ungermann-96]) a detailed codification of the company's processes, a process model in some kind of notation.

Time to Market The faster time to market for business processes will force the same pressure on the software development. This will make an integration of the various sub-processes a must.

Quality of reusable components Only high quality components will in fact be reused.

Emphasis on SMEs The software industry, especially in Europe, is characterized by small to very small companies (in Austria 50 % of the companies in the software industry have less

than 5 employees, 65% have less than 25 employees [Hofer-96a], cf. Fig. 4). The European scene shows a similar picture. If the call for 'industrial software development' and the quest for certification a la ISO9000 or ISO 15504 is to be taken serious we need the availability of low-cost, easy to use integrated CASE-Tools Environments integrated with an appropriate Process Model and an interface to a cheap, easy to handle process management tool. It should also allow persons not well skilled in project management tools to make plans and to judge the consequences [Ebner-95, Hofer-96, Hubert-90]. The message has been understood by the European Commission which in increasing amounts launches programs for small and very small software developing units (e.g. ESPITI [Chroust-96l, Chroust-96g]), SPIRE) and supports standardization/improvement efforts like SPICE (now draft ISO 14405 [Simon-96]).

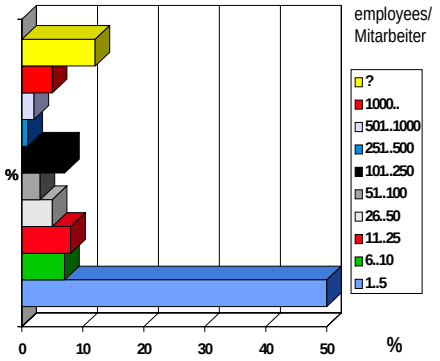

Figure 4: Distribution of employees in the Austrian Software Industry

5 What are the show stoppers?

With all the advantages of both process-oriented Software Engineering Environments and Integrated CASE-Tool Environments one wonder, why we find to little acceptance. We will look at it from the viewpoint of a *small software developer*, an 'SSD' in the SPIRE terminology. Several show stoppers can be identified

cost Environment of the described type run easily into an order of 100.000$ and more - a sum outside of the reach of a small 4 person software shop. This is due not only to the actual cost (several ten thousand dollars) but also due to the manpower need to maintain such systems ('cost of ownership').

size, complexity, maintenance Such environments have to be maintained, this takes considerable know-how and manpower. In the SSD there simply not enough man power to cover this.

Big Bang At least the introduction of a process-centered Software Engineering environment has to be done at once - at least in a pilot project. With the small size of such companies the pilot project constitutes a considerable portion of the whole business.

capability level The definition of capability levels [Paulk-95] indicates that only at *level 3* a defined process models is effective Fig. 5 Since most software development companies

are between level 1 and 2 process centered software engineering environments seem to be premature.

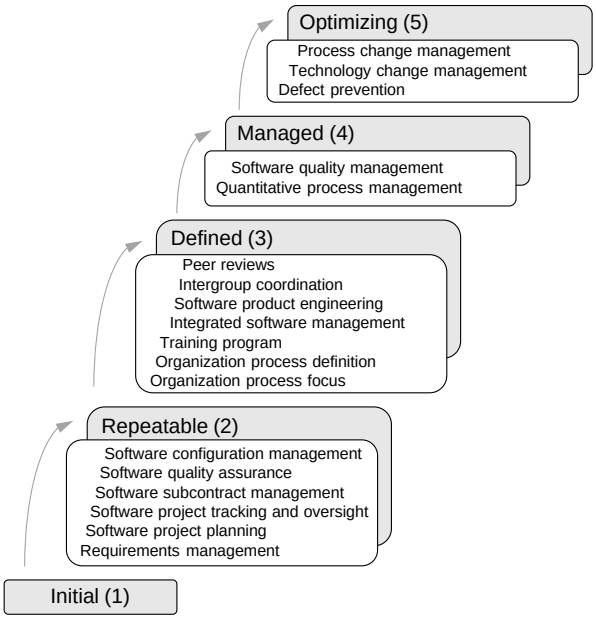

Figure 5: Capability Levels and Key Process Areas

cost of technology inhomogeneity It is known [Hoch-91] that changing to a new technology involves additional effort and/or loss of productivity. Many SSDs simply cannot afford either.

danger of bureaucracy Especially very small software house work in a very informal way, each employee performing various and varying jobs, a process model seems to disturb this and to force additional paperwork of information which usually rests in the heads of a few.

single-mindedness of software engineering environments If one looks at the time distribution of a typical software engineer's time one sees that only a rather small portion (50% of a typical team leader seems to be communication). This means that the support of a process model is not such a big help as supposed. Secondly the activities besides direct technical software engineering are not supported and modelled at all.

Above show stoppers have several consequences:

- For small companies it is cheaper to ignore process models and rather 'learn' the process by heart.
- Instead of buying an integrated CASE-Tool Environment a few tools are bought. Bridges between them provide the information transfer.
- The previous item also makes a Repository unnecessary, its price would have been prohibitive anyway.

- Not having any consistent data on the process progress a simple, stand-alone project management tool suffices for basic project management.

6 What are the challenges?

The fact that above strategies are forced by the economic and not by technical factors should be a challenge to the 'meta-software industry' to produce more adequate tools for the very small market and not get fascinated by the few big spenders. The main challenges seem to be:

- Create small, low-cost, easy-to-maintain tools for small companies and
- provide integrated environments binding together process and project management and tool support, including a repository.
- provide enough process data collection that reasonable project management tools can be attached and provided with adequate data.

If the push for process certification continues and the issue of quality will become more important, small companies will have to get accustomed to the idea of instrumenting their software development business. And this is a rather large market share. Therefore it is the area of the small enterprises were the market (and the money) of the future lies ...

References

[Bailetti-95] Bailetti, A.J. FitzGibbon, C. *ISO 9001 Registration for Small and Medium-Sized Software Enterprises* Carleton Univ. Press ISBN 0–88629–255–7.

[BAInf-95] Bundesamt, f. Informatik (ed.) *HERMES – Führung und Abwicklung von Informatikprojekten* Bundesamt f. Informatik, Schweizerische Bundesverwaltung, Bern, Ausgabe.

[Bandat-90] Bandat, K. *Process and Project Management in AD/Cycle* Proc. 31st GUIDE Spring Conference, Bordeaux, June, pp. 55–60.

[Berkeley-90] P., Berkeley D. De Hoog R. Humphreys *Software Development Project Management: Process and Support* Ellis Horwood, New York.

[Broehl-92] Bröhl, A. P. Dröschel, W. (eds) *Das Vorgehensmodell in der Anwendungsentwicklung – Standard und Leitfaden* Oldenbourg München.

[Broehl-93] Bröhl, A.P. Dröschel, W. (eds) *Das V-Modell – Der Standard für die Softwareentwicklung mit Praxisleitfaden* Oldenbourg ISBN 3–486–22207–4.

[Chroust-88g] Chroust, G., O. Gschwandtner and , D. Mutschmann-Sanchez *Das Entwicklungssystem ADPS der IBM* Gutzwiller T., Österle H. (eds.): Anleitung zu einer praxisorientierten Software-Entwicklungsumgebung, Band 2 AIT Verlag München, pp. 123–148.

[Chroust-89d] Chroust, G. *Application Development Project Support (ADPS) – An Environment for Industrial Application Development* ACM Software Engineering Notes, 14:5:83–104.

[Chroust-90b] Chroust, G., H. Goldmann and , O. Gschwandtner *The Role of Work Management in application development* IBM System Journal, 29:2:189–208.

[Chroust-91a] Chroust, G. and , S. Knotter *Vom phasen-orientierten zum task-orientierten Vorgehen in Informatik-Projekten* Elzer P. (ed.): Multidimensionales Software-Projektmanagement AIT-Verlag München, pp. 81–111.

[Chroust-91e] Chroust, G., G. Heger and , P. Pfann *Integrating the Use of AD/Cycle Tools under ADPS (A Restructuring Environment)* Proc. SHARE Lausanne, April, pp. 15–27.

[Chroust-92a] CHROUST, G. *Modelle der Software-Entwicklung – Aufbau und Interpretation von Vorgehensmodellen* Oldenbourg Verlag.

[Chroust-92g] CHROUST, G. *Computer Integrated Work Management (CIW)* Mittermeir R. (ed.): Shifting Paradigms in Software Engineering Springer Verlag Wien New York, pp. 4–13.

[Chroust-93h] KITZMÜLLER, K. and , G. CHROUST *Navigation and Activity Networks* Sima D., Haring G. (eds.): The Challenge of Networking – Connecting Equipment, Humans, Institutions, Proc CON'93, Szombathely, Oldenbourg Wien München, pp. 83–96.

[Chroust-94b] CHROUST, G. *Navigation in Process Models* Warboys B.C. (ed.): Software Process Technology – 3rd European Workshop EWSPT'94, Villard-de-Lans, Springer Lecture Notes No. 772, Feb 94, pp. 119–122.

[Chroust-94d] CHROUST, G. and , S. HOFER *ADW-Ausbildung – geht es auch ohne?* Knowledge Ware (ed.): Austrian User Conference, 17 18. Mai, Wien.

[Chroust-96g] CHROUST, G., (ed.) *Special Issue: ESPITI - Guest Editorial.*

[Chroust-96l] CHROUST, G. *Why does the European Community support Software Process Improvement?* Science and Technology, no. 2, April, pp. 8.

[Corzilius-92] CORZILIUS, R. (ED.) *AD/Cycle – Ziele, Konzepte und Funktionen* Oldenbourg-Verlag München Wien.

[CTN-95] CTN *PROSIMULA - Teilnehmer Handout.*

[Deltl-89] DELTL, M. *MERISE – Theorie und Implementierung einer Analyse- und Entwurfsmethode* Diplomarbeit, Techn. Univ. Wien.

[Dorling-93] DORLING, A. *SPICE – Software process improvement and capability determination. Information and* Software Technology(35) p.404 (June).

[Downs-88] DOWNS, E. CLARE, P. COE I. *Structured Systems Analysis and Design Method* Prentice Hall, Englewood Cliffs SSADM.

[Dowson-86b] DOWSON, M. (ED.) *Iteration in the Software Process 3rd International Software Process Workshop, Breckenridge Colorado* Nov. Participants' Proceedings.

[Dowson-91a] M., DOWSON B. NEJMEH W. RIDDLE. *Fundamental Software Process Concepts* Fuggetta A., Conradi R., Ambriola V. (eds) Proceedings of First European Workshop on Software Process Modeling, CEFRIEL, Mailand, Italy, May, AICA, pp. 16–38.

[Dowson-94] DOWSON, M. FERNSTRÖM, C. *Towards Requirements for Enactment Mechanisms* Warboys B.C. (ed.): Software Process Technology – 3rd European Workshop EWSPT'94, Villard-de-Lans, Springer LNCS 772, Springer Berlin-Heidelberg„ pp. 90–106 ISBN 3–540–57739–4.

[Ebner-95] EBNER, M. ROSSOW R. *Project Simulation Prosimula* in: *IDIMT '95 - 3rd Interdisciplinary Information Management Talks*, pp. 120–124, München Wien Oldenbourg.

[Garra-95] GARRA, A. *Vorgehensmodelle mit GIK/2* Diplomarbeit Kepler Universität Linz, Inst. f. Systemwissenschaften Mai.

[Habermann-93] HABERMANN, H.J. LEYMANN, F. *Repository – Eine Einführung* Oldenbourg ISBN 3–486–22200–7, 290pp.

[Hares-90] HARES, J.S. *SSADM for the Advanced Practicioner* John Wiley Chichester-New York ISBN 0–471–92739–2.

[Hausen-81a] HAUSEN, H.L. MÜLLERBURG M. *Conspectus of Software Engineering Environments* Proc. 5th Conf. on Software Engineering, pp. 34–42.

[Hausen-82] HAUSEN, H.L. MÜLLERBURG M. *Software Engineering Environments: State of the Art, Problems and Perspectives* Proc. Compsac '82, Nov., pp. 326–335.

[Hausen-85] HAUSEN, H.L. MÜLLERBURG M. SNEED H.M. *Software-Produktionsumgebungen* R. Müller Köln.

[Hazeyama-92] HAZEYAMA, A. and , S. KOMIYA *A Process Model for Software Project Management* Annual Review in Automatic Programming, 16(Part II: Proceedings of the Fourth IFAC/IFIP Workshop):109–116.

[Hoch-91] HOCH, D.J. *Management von Technologie-Diskontinuitäten in Informatik Projekten* Elzer P. (ed.): Multidimensionales Software-Projektmanagement AIT-Verlag München, pp. 153–179.

[Hofer-95] HOFER, S. *Importance of Project Management in Software Development* in: *IDIMT '95 - 3rd Interdisciplinary Information Management Talks*, pp. 108–119, München Wien Oldenbourg.

[Hofer-96] HOFER, S. *Project Management Expertise Through Simulation* Cybernetics and Systems: An International Journal, 27(2):169–182.

[Hofer-96a] HOFER, S. *Software-Entwicklung in Klein- und Mittelbetrieben: Situation, Probleme, Möglichkeiten* Techn. Report

[Hubert-90] BRASSEUR, L. HUBERT F. FOURNIER M. BOURDON-LE *Eureka Software Factory: OPIUM An environment for Software Process Modeling integrated with a Project Management Tool* in: *6th International Software Process Workshop*, pp. 103–107 IEEE Computer Society Press.

[ISO-12207] ISO, (ED.) *ISO-12207, Information Technology, Life Cycle Processes* Int. Org. for Standarization, ISO 12207.

[ISO-94] ISO, (ED.) *ISO 9000–3: Guidelines for the application of ISO 9001 to the Development, Supply and Maintenance of Software* Int. Org. for Standarization, ISO 9001, June 1.

[Kay-90] KAY, S. *The No. 1 Life Cycle Method* Computer World, 24:34, p 82.

[Kehoe-96] KEHOE, R. JARVIS, A. *ISO 9000–3 – A Tool for Software Product and Process Improvement* Springer New York ISBN 0–387–94568–7.

[Martin-89a] MARTIN, J. *Information Engineering, Book I: Introduction* Prentice Hall, Englewood Cliffs.

[Merbeth-89] MERBETH, G. *MAESTRO-IPSE – die Integrierte Software-Produktions-Umgebung von Softlab* Balzert H. (ed.): CASE – Systeme und Werkzeuge B-I Wissenschaftsverlag, pp. 213–234.

[Mercurio-90] MERCURIO, V.J. MEYERS, B.F. NISBET A.M. RADIN G. *AD/Cycle strategy and architecture* IBM System Journal, 29:2:170–188.

[Paulk-95] PAULK, M.C. *The Evolution of the SEI's Capability Maturity Model for Software* Software Process, pilot issue August, pp. 3–15.

[Privateer-90] PRIVATEER, P. *AD/Cycle: KnowledgeWare's Perspective).* Proc. 31st GUIDE Spring Conference, Bordeaux, June, pp. 28–46.

[Rout-95] ROUT, T.P. *SPICE: A Framework for Software Process Assessment* Software Process, pilot issue August, pp. 57–66.

[Sagawa-90] SAGAWA, J.M. *Repository Manager Technology* IBM System Journal, 29:2:209–227.

[Simon-96] SIMON, J.M. *SPICE – Overview on Software Software Process Improvement* Special Issue on ESPITI (European Software Process Improvement Training Initiative) Journal of Systems Architecture.

[Ungermann-96] UNGERMANN, C. TESCH, F.J. STOLPE B. WEISSERT M. *Qualitätsmanagement bei der Software-Erstellung* VDI-Verlag Dsseldorf 1996.

[Yourdon-90] YOURDON, E. *Softlab's Maestro* American Programmer, 3:3.