Basic Support for Intranet Software Processes

Hans-Ludwig Hausen
GMD-FIT-CSCW, Schloss Birlinghoven, D-53757 St.Augustin Germany
<> www.scope.gmd.de <> hausen@gmd.de <>

1997-02-20

Abstract

In order to support software project groups over the internet one needs to have at least four basic instruments: a software process programming and testing tool, a programmable generator to create container and templates for software artefacts, a team information coordination tool to control access to software artefacts and finally a software artefact information system. The process support tool has to be applicable to define and test selected attributes of the the process. Based on this assumptions we investigated whether it is possible to use ProcePT, VMbrowser, BSCW and a Data Base System to create an basic intranet infrastructure for collaborative software process engineering. The conclusion is a stepwise approach seems possible. The first step is to use the instruments in its own environment and establish manually communication between them by transferring files. Incremental enhancement will lead to a integrated set of small tools to support process programming and testing by a modified ProcePT, artefact template generation by a set of word processing macros, artefact management by a data base system respective by an object information system, and collaboration management.

1 Motivation

Software engineering today is shifting from traditional work environments which provide the set of classical methods and tools to distributed and asynchron developments. This requires on the other hand a solid definition of the total software process. Such software processes have to generic in order to be applicable in different situations and environments and such process have to be adaptable in order to allow application of dedicated methods and tools. To manage such processes in a distributed environment on the other hand might be instrumented by data bases and workflow instruments. As such instruments are often highly integrated and costly a baseline for netbased cooperative software engineering has been investigated. The aim was to provide basic methods for software process planing and controlling and to instrument such processes by methods for group interaction. Requirement was to allow the application of state of practise text processing instruments.

2 Software Process Programming and Testing

In order to ensure the quality of a software product and in particular the productivity of the processes of system development and application, the software process has to be specified and tested. Starting from a generic process model, activities and documents are

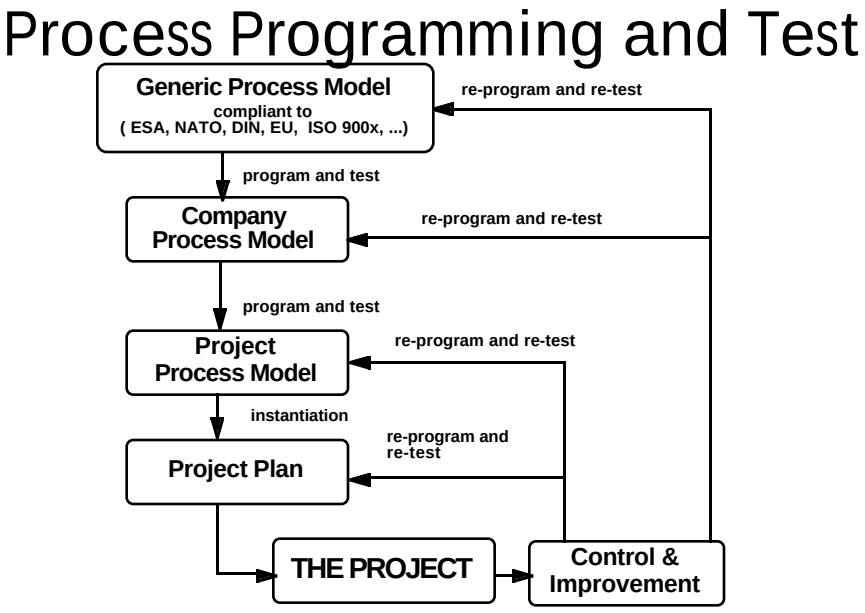
tailored until the project-specific process model is achieved. Adaptation normally takes place stepwise depending on the specific conditions considered. The GMD approach ProcePT (Process Programming and Testing) supports these steps.

The approach allows the specification and testing of specialized process models by adaptation of the German process model "Vorgehensmodell" (V-Model). The V-Model is required by the German government, and its use is widespread in the German IT industry. It has also influenced EUROMETHOD, the European integration project for software process models.

The V-Model names the documents and activities relevant for software development on different abstraction levels. Furthermore, it distinguishes between different sub-models: software development, project management, quality assurance, and configuration management. Such a process description requires among other things formal notions for refinement and generalisation. A rule-based approach has already been developed and implemented within an prototype. Adaptations to other process models like IDEFx, SSADM or MERISE are possible.

Following a rule-based approach according [**?**] and [**?**] the ProcePT prototype (Prolog program of about 3500 rules) [**?**]offers online documentation of the V-Model and tailoring procedures. The guided tailoring follows the so-called deletion conditions. Further documents and activities can be identified for deletion by unguided tailoring. Then the current set of facts is tailored according to the deletion conditions identified. In this transformation several tests are undertaken:

<> testing the document and activity flow,
<> testing the status changes,
<> testing the interface between the submodels,
<> observing the level-oriented document and activity structure,
<> avoiding isolated documents and activities.



The adaptation procedures as well as the conformity testing to standards like ISO 9001 or inhouse-standards, are reported in a protocol that is needed for process audit. The data of a specified model are made available by an SQL-interface (Structured Query Language). ProcePT supports the creating of a project handbook. Further aspects of project management, for example the planning and testing of resources, might be integrated soon.

ProcePT provides support for:

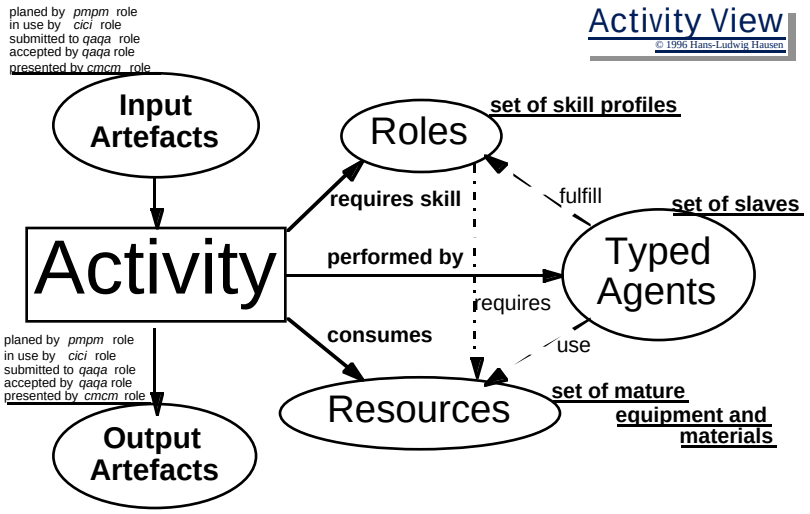
<> software project managers

<> clients and employers who determine the software development procedure

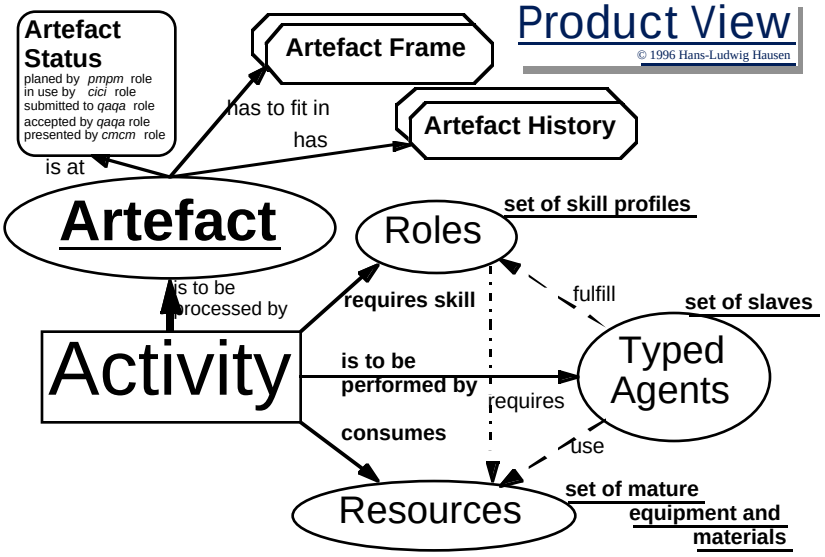
<> developers of IT systems who need to equalize the procedures used

and requested * test houses and quality assurance groups who need to test the compatibility of software process models and to conduct process certification.

Form a software infrastructure management point one might distinguish a product view and a activity view.



One might argue that the activity view provides an appropriate representation and overview of the process whereas the product view will help to identify from a standpoint of an individual what type of actions on which products have to be conducted. The first approach therefore would be to identify all artefacts belonging to a certain agent and then to attach to each artefact the activities to be performed on a particular artefact.



With this assumption in mind we need from our ProcePt tool for each for a given process for each particular agent (at least the list and source reference of)

- artefacts handled,
- activities to be conducted on each artefact,
- agents and roles active in the process,
- product template, and
- product flows

as the essential parts of an interface to any supporting instrument.

As indicated above we don't require a particular organizational structure, thus allowing a project manager to introduce a structure she/he feel to be appropriate.

3 Access and Collaboration Support provided by BSCW

This section looks briefly at those features of the present BSCW (Basic Support for Cooperative Work) Shared Workspace Server which is to be used to support the collaboration of work groups with software process engineering.

According to [**?**] and [**?**] the BSCW Shared Workspace Server provides in addition to the features of today's Web servers facilities for collaborative work on artefacts such as:

- shared workspaces offering features to handle a fixed set of typed artefacts (folders, documents, spreadsheets, pictures or links, etc.),
- features for uploading (storing into the workspaces) from and downloading (obtaining from the workspaces) of artefacts to their local computing platform (their local file system),
- user's workspace membership management (administration, authentication and authorization) via user names and passwords,
- artefact version management via a so called 'soft' locking scheme that reports time and user who performs a so-called check out and keeps versions till their explicit removal by an authorized user,
- monitoring, notification and awareness service for events occurring within a workspace.

In its present form BSCW does not provide features for

- active notification of users within a workspace and of users within other workspaces ,
- mixed asynchronous and synchronous cooperation, or even for total synchronous collaboration,
- direct notification and/or observation (after permit) of the presence of workspace members
- direct grouping of users within workspaces or across workspaces
- programmable and default artefact transmission

- semantic status information for folders and documents (e.g. in use by, submitted to, approved by, ..)
- artefact modeling for a workspace allowing the definition and management of relations between artefact, activities to be conducted on it, roles of those involved, resources consumed, agents actually in charge for an activity on an artefact,

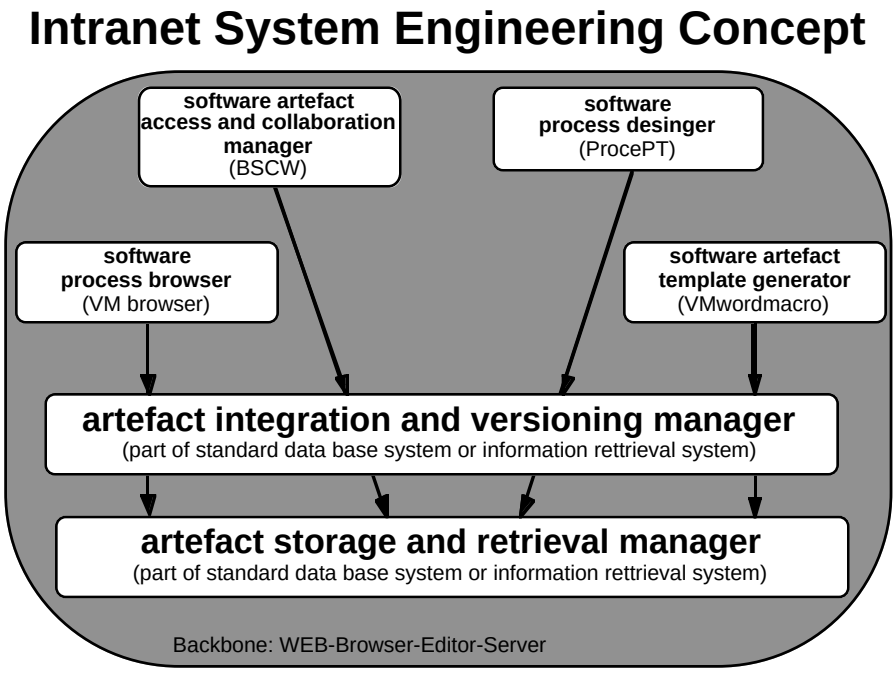
It has been shown in the previous sections that these feature will be useful in the support of collaborative software process engineering. These feature should be of the whishlist for further enhancement of BSCW.

4 Components of an Intranet Software Process Instrumentation

If software projects should be supported over the net the following components might be required

- a software process designer (e.g. ProcePT, c.f. www.scope.gmd.de),
- a software process browser (e.g. VMbrowser, c.f. www.scope.gmd.de),
- a software artefact template generator (e.g. VMwordmacro, c.f. www.scope.gmd.de),
- a software artefact access and collaboration manager (e.g. BSCW, c.f. bscw.scope.gmd.de).

As we learned in the debate on software engineering environments, complex general problem solvers are not accepted, because of their inflexibility and unjustifiable cost. One must require basic features for each of the items named above.



As indicated in the picture above we assume a separation of concerns. The

- software process designer produces the static process lay out indicating the objects, activities, roles, resource, agents and their static relation ships,
- software process browser allows at every point in time the retrieval of process information (e.g. product flow, activity description, conduct of work description, etc.),

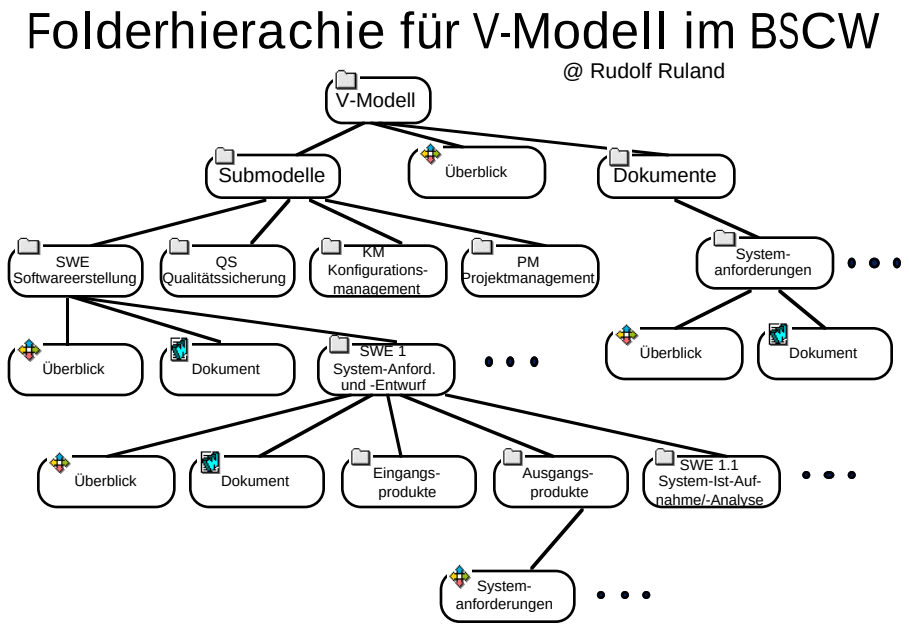
- software artefact template generator produces the respective templates upon request of the particular agent,
- software artefact access and collaboration manager defines the intranet working environment for each agent.

5 CSCW for Process Engineering

With respect to the features and aims of BSCW there are two way to use it effectively for software system process engineering: Editing a Process Model and Controlling a Software Process.

BSCW within Editing a Process Model

For shared editing of a software process model one has to represent the model using the BSCW data modeling features. At present that leads to a representation as follows:



Within the V-Model editing context

- a shared workspace is proposed to be used for each submodel (SD, QA, CM, and PM). Nested folders are to be used in particular to represent the decomposition tree of the model into submodels attached additional documentation,
- uploading and downloading of submodels between the reference model (at the ANSTAND-office) and the contributors local computing platform (i.e. their local file system) is to be controlled by the editor of a particular submodel,
- user's workspace membership management (administration, authentication and authorization) is conducted by the submodel editor via user names and passwords,
- submodel version management is performed by default via time stamp and user identification,
- monitoring and notification is attached to the submodel editor.

At this point one might want to enhance this type of intranet editing by a full relational representation of the process model thereby allowing an immediate consistency check. The question is whether one can have structured documents as proposed by an underlying information system controlled transmitted in both directions. In other words, can the functionalities of a data base system made available at the workspace level for each member in a dedicated, proper fashion.

BSCW within Controlling a Real Life Software Process

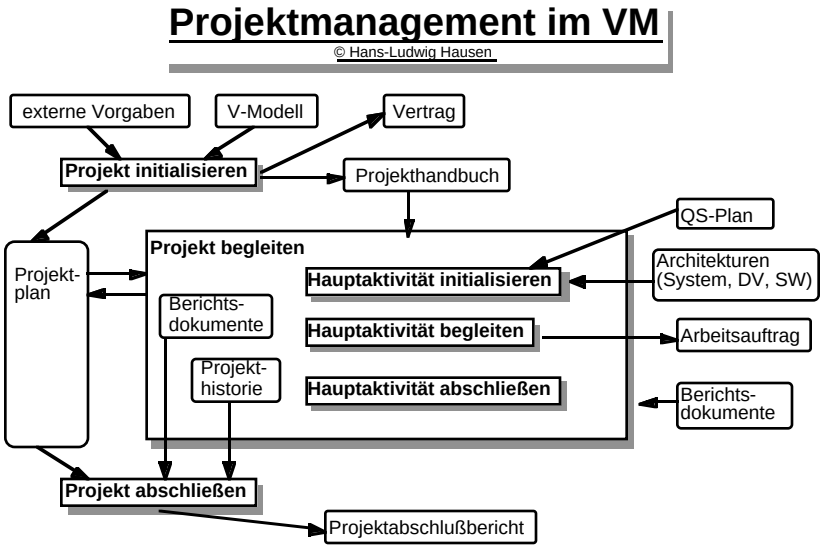
As discussed above we got a small number of typical process models: phase, waterfall, v-shape and spiral. If we look at them from a product point of view we just can use the BSCW workspace concept to manage access and collaboration.

Product Viewpoint

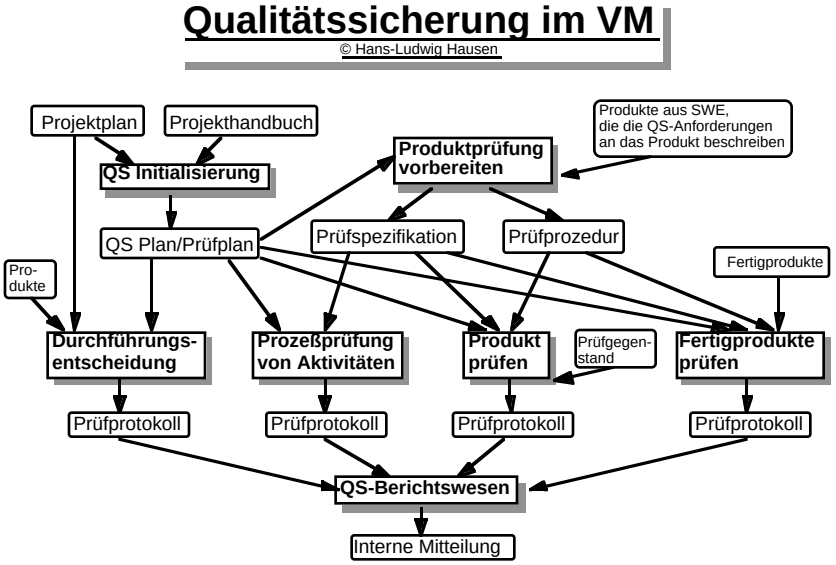
The product point of view leads to a definition and control of

- composed artefacts and related composed activities by folders
- artefacts and related activities by documents
- agent identification by workspace member identification
- artefact status via documents
- artefact templates via documents
- artefact history via documents
- activities via folders containing descriptions of role and resource consumption of the workspace member attached to that particular task for the indicated role.

The access and collaboration actions are conducted according the project management tasks within workspaces for the particular project management artefacts.



A second example of schematic activity is given for quality assurance. As for development and for configuration management transfer is indicated by activity object relation.



Administration, authentication and authorization as well as monitoring, notification and awareness of events occurring within a workspace are pre-defined with the PM and QS actions. In order to proceed in a process PM or QS create or delete workspaces and define user and workspace membership according their active workspaces.

In order to provide awareness on the substance of work a member of a workspace has to have access to the project manual (Projekt Handbuch)

The interaction between BSCW and ProcePT requires the transformation of ProcePT output into workspace templates for each artefact. Such workshop templates can be generated by using the

- agent to role designation,
- product flow description,
- project manual

from the ProcePT output interface. Description of artefacts and activities are to be obtained by hypertext link form the VMbrowser. The product templates are generated and integrated into the workspace versioning system.

Active notification of users within a workspace and of users within other workspaces would be very useful for the PM to trigger the process. As both asynchronous and synchronous collaboration might occur in a workspace both features would be appreciated. Direct notification and observation of workspace members is not mandatory by the V-Model. Nesting and coupling of workspaces would make possible a very effective realisation of intranet process engineering.

6 Conclusions

The discussion of today's principal process models reveals a basic structure of such processes: collections of partially connected networks of activities defined by flow of software artefacts. In order to support such processes one needs to have support to program and test a process, to allocate and provide access to software artefacts to agents (designers, programmers, testers, ..., managers), to outline software artefacts, and to configure, store and retrieve software artefacts.

For the management of software we can make use of any standardised software artefact data base which provides configuration management features. Using then basic instruments such as ProcePT for software process programming, BSCW for web-based computer supported cooperative work, VMbrowser for viewing process models, and an artefact frame generator (e.g. a macro set for a standard word processor) one gets a basic platform for intranet based software process engineering.

References

[Back96] Bcker, A., Busbach, U., DocMan: A Document Management System for Cooperation Support, in Proceedings of the 29th Hawaiian International Conference on System Sciences, Maui, Hawaii, IEEE Computer Society Press, January 1996, pp. 82-91

[Bent95] Bentley, R., Horstmann, T., Sikkel, K. and Trevor, J., Supporting collaborative information sharing with the World Wide Web: The BSCW Shared Workspace system, in Proceedings of the 4th International WWW conference, Boston, O'Reilly, December 1995, pp 63-74

[Haus89] Hausen, H.L. *Rule-Based Handling of Software Quality and Productivity Models* in: C. Ghezzi J.A. McDermind (Eds.) ESEC'89 2nd European Software Engineering Conference 1989, University of Warwick, Coventry, UK, September 1989, LNCS 387, Springer Verlag, Berlin, p. 376-394.

[Haus92] Hausen, Hans-Ludwig: A Specification of an Assessment and Certification Advisor, in: Annual Conference of the ACM, March 1992, Kansas City, MO, 20 pp

[WHS95] Welzel, Dieter; Hausen, Hans-Ludwig; Schmidt Walter: Tailoring and Conformance Testing of Software Process, in: Proceedings of the Second IEEE International Software Engineering Standards Symposium (ISESS'95), Montral, Quebec, Canada, August 21-25, 1995, IEEE Computer Society Press,1995 and enhanced version in: Proceedings of the fifth International Converence on Software Quality (5ICSC), Austin, Texas, October 23-26, 1995, pp. 409-414, ASQC Software Division, 1995.