The Influence of Environmental Parameters on the Acceptance of Collaboration Tools

Astrid Behm, Asarnusch Rashid

FZI Forschungszentrum Informatik
Haid-und-Neu Strasse 10-14
76131 Karlsruhe
behm@fzi.de, rashid@fzi.de

Abstract: The configuration of the collaboration environment has a significant influence on the development and the use of a collaboration tool. During the development process an optimal adaptation of the collaboration tool to the environment takes place. This has a substantial effect on the success and the use of the tool which depends on the acceptance by the end-users. The use of a collaboration tool in a new, different collaboration environment depends on the match of this environment with the environment the tool was developed in. This has to be considered and checked during the selection process of collaboration tools. However current selection procedures for software do not comprise any criteria supporting the comparison of collaboration environments. The current article submits a proposal for an add-on to selection procedures defining new selection criteria. The deduction of such criteria is presented for the sub process of 'Elicitation of problem and change requests' of the software development process on base of the collaboration tool OpenProposal.

1 Introduction and Problem Description

Software development is a complex process with intensive mechanisms in communication, information exchange and decision-making. Increasingly the process takes place both geographically distributed and in an inter-organisational mode. It is only partially automatable and considerably influenced by its social component. The increasing importance of collaboration in the process of software development is reflected in the research of the recent years. CSCW research focuses on the development of tools and techniques for the assistance of collaborative processes as well as the psychological, social and organisational influences on the software development process itself [Gr94]. A broad agreement exists that the use of tools supporting the collaboration in the software development process is vital for the production of high-quality software [Al99, Co04]. The diversified field of collaboration tools span complex software development environments with embedded collaboration functions as well as special tools only focusing a concrete single collaborative sub-process. The product market of collaboration technologies is growing fast with as many as 1000 vendors that offer software products with collaborative functionality [Co05].

The success of application software and collaboration software are subject to different conditions. Application software is result-oriented. After having decided to use a concrete application system in an organisation no other way of executing the task will be possible in future. Besides the completion of the requirements of usability and infrastructure and the supply of interfaces first of all the success of the application software depends on the quality of the realisation of the functional requirements relevant for the execution of the task defined by the stakeholders. Collaboration tools are not result-oriented but focus on the manner a task is executed. The result of the task does not depend on the use of a concrete system. So collaboration tools support the user by the performance of tacit, hidden and amorphous processes [Eh99]. Following examples serve as illustration of this peculiar characteristic.

- Communication can take place by mail or by phone. The user is not forced to use the mailing-system.
- Requirements elicited in the context of the requirement engineering process can be sent to the analysts in the form of word documents or excel spreadsheets as well as by the use of a special collaboration tool designed for the support of this process.
- The success of a shared calendar depends on several factors. The alternative of the use of a shared calendar is not 'using a different tool', but not publishing any date.
    - Is the privacy of certain dates respected?
    - Do the collaboration partners trust each other? Otherwise they will not publish their business dates especially not if the collaboration takes place cross-company.
    - Can the shared calendar be synchronised with mobile computing units?

First of all the aim of the use of a collaboration tool is the support of the creation and transfer of information. Collaboration tools are invasive, not only bias the user's operating principles but have significant influence on the communication behaviour of the user. Whether a collaboration tool is used in the intended manner depends not only on the disposable functionalities but also on the acceptance of the tool by the users [BOO95, Te00]. The influence of the environmental conditions that are external, non-functional aspects on the use, the acceptance and the success of a collaboration tool [CO07, HM03, Th07] is beyond question.

If the implementation of the application software meets the functional and technical requirements it can be assumed that the adoption of the application system in an organisation will be successful. As stated above, the success of a collaboration tool depends to a great extent on the acceptance of the tool by the user. In order to rate the expected acceptance of the collaboration tool by the user in a certain collaboration environment these environmental conditions have to be listed concretely. For this purpose a formal approach cannot be found in the literature. This paper takes up this challenge. During the project CollaBaWü aiming among other the development of tools supporting the collaboration in the software development process the system OpenProposal was developed. OpenProposal is specialised on the support of the sub-process 'Elicitation of Problem and Change Requests' of the software development process. Accompanying the process of evaluating OpenProposal a list of external parameters influencing the acceptance of OpenProposal was written down. The description of these environmental parameters and their deduction from the evaluation results of OpenProposal are presented in this paper.

Statements about an expected acceptance of a collaboration tool can be made by matching the environmental parameters of the situation the collaboration tool was developed for and the environmental parameters of the situation the collaboration tool is intended to be used in. Supposing the collaboration tool is meeting all the functional and technical requirements defined for the software system statements concerning the expected acceptance of the tool in the new collaboration environment can be made based on the relation between the values of the environmental parameters. If the values of the relevant environmental parameters correspond a possible success and acceptance of the collaboration tool in the new collaboration situation cannot be predicted. But if there are significant divergences on substantial aspects of the collaboration environment a statement can be made that there will be no success and acceptance for the collaboration tool in the new environment. As the value of many of the environmental parameters can be influenced by the organisation, activities for the introduction phase of the collaboration tool can be deduced from the divergences and values of the environmental parameters.

The rest of the paper is organised as follows. Firstly some related work is presented. Secondly the substantial differences between the situation of the development and the situation of the selection of a collaboration tool concerning the environmental parameters and the different relevance of environmental parameters in diverse processes will be worked out. Thirdly the relevant environmental parameters of the sub process ‘Elicitation of Problem and Change Requests’ will be derived from the evaluation results of OpenProposal.

2 Related Work

Selection procedures of software systems are based on a list of criteria including a description of the objectives, the functional requirements of the software system and the definition of the minimum requirements [ex. Bu07a, Bu07b] as well as indications of the expected quality of the software product (cf. ISO/IEC 9126, DIN 66272). On top of that the list comprises a description of the IS infrastructure the software system is expected to work in [Gr01, SH05]. [Gr01] criticises the intense focus on functional decision criteria in existing well known selection procedures for standard software systems. Several proposals for non-functional extensions to the criteria used by these selection procedures are made [CFQ07, Te02]. Nevertheless as special tools need special selection procedures taking their peculiar features into account [Te02] these procedures are approaching their limits [We06]. So a selection procedure for collaboration tool has to consider the characteristics making up collaboration tools.

Since its inception the classification of collaboration tools has been a substantial focus of the CSCW Research area. The time-space taxonomy by Johansen in 1988 or the functional classification [BS00, EGR91] contributed to a fundamental structuring of collaboration tools and worked out the major categories of applications and technologies.

Due to the continuous integration of a variety of functions and technologies in products supporting collaboration these classification schemes do no longer meet the requirements to describe a collaboration tool in full, especially since they do not take into account the particularities of the non-result-oriented character of a collaboration tool. Several theories such as the media richness theory [DL86], the coordination theory [MC94] and the task technology fit theory [ZB98] contributed to a better understanding of the characteristics of collaboration processes, tasks and technology. Because of these theories being “either too general in nature not offering sufficient granularity for a detailed requirement analysis and evaluation of alternative solutions, or focus(ing) only on parts of the ‘big picture’” [We06], they offer only limited practical guidance for the selection of collaboration tools. Lessons learned by these theories have to be integrated into a superior framework of environmental parameters and their mutual dependencies. Facing the problem of a large portfolio of collaboration tools characterised by numerous independent tools for communication and information sharing within a large Norwegian based oil and gas company the authors of [We06] were forced to develop a method for validating this portfolio. Hence they developed a broad framework consisting of collaboration process, environment and support describing all aspects a selection procedure for collaboration tools has to consider. The details of this framework described in the article were limited to the covering of the description of the collaboration process of the framework. For further research [We06] emphasises the necessity of a formal framework describing the collaboration environment enabling the characterisation of the political, economical and organisational factors influencing the collaboration on the level of the organisation, the team and the individual. [CFQ07] presented an extension of formal lists of criteria consisting of non-functional aspects. As these criteria are generally applicable on selection procedures for any kind of software they do not meet the particularities characterising collaboration tools. Additional criteria are needed regarding the dependency of the success of a collaboration tool from the acceptance of the tool by the user. On reflection of these criteria it should be possible to make a statement concerning the expectation of the acceptance of a collaboration tool.

3 Environmental Influences on Collaboration Tools

As stated above, collaboration tools are not result-oriented but support-oriented. Hence the development process of collaboration tools follows different conditions during the process of requirement engineering and design than the development process of standard application systems. [Pa01] underlines the fit between the task, the technology and the users which has to be considered during the development, others stress the participation of the users on the software development especially in case of developing collaboration tools [HL01, JMS02]. Collaboration tools are designated to be part of a social system and the development process has to address the issues of this social system [An03]. The acceptance of a collaboration tool depends on many underlying conditions describing the collaboration environment to which attention has to be paid already during the development process of the collaboration tool. Social awareness [BH04, CNH04, GPS4, TGG06], the existence of personal and social networks [HM03], team processes and the existence of group basis [KL99] and privacy [FLP06, HI05, RI06] are some of them. Not every condition can be mapped into functional requirements of the tool. So these remaining underlying conditions have to be created and maintained outside the tool. In Figure 1 the situation during the development of a collaboration tool is illustrated.

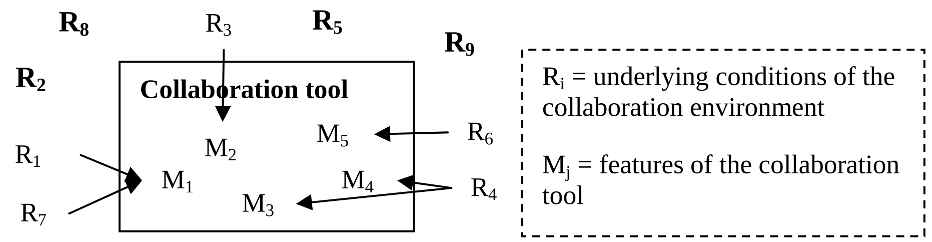

Figure 1: Mapping of the underlying conditions of the collaboration onto the features of a collaboration tool

Communication modalities, like the persistence of information or the facility of feedback, for example, can entirely be mapped onto features of a tool. Different underlying conditions like the need of social awareness can only be partially mapped onto a feature - for example by integrating a component representing this underlying condition. How far this component can fulfil the expectations of each individual depends on the given case and implies the component being designed configurable. Other underlying conditions like the existence and the control of the adherence of group basis or trust can not be represented by features of a collaboration tool (R_2, R_5, R_8 und R_9 in figure 1). But they have tacit influence on the design of the collaboration tool during the development process.

[Pa01] points out that the findings describing the issues of one social system can not just be transferred into another social system. So the developed collaboration tool just fits the initial collaboration situation and has to be adapted and improved for the use in another situation. In other words, if the collaboration tool is to be used in a different environment for example in another project or another organisation the collaboration environment will change.

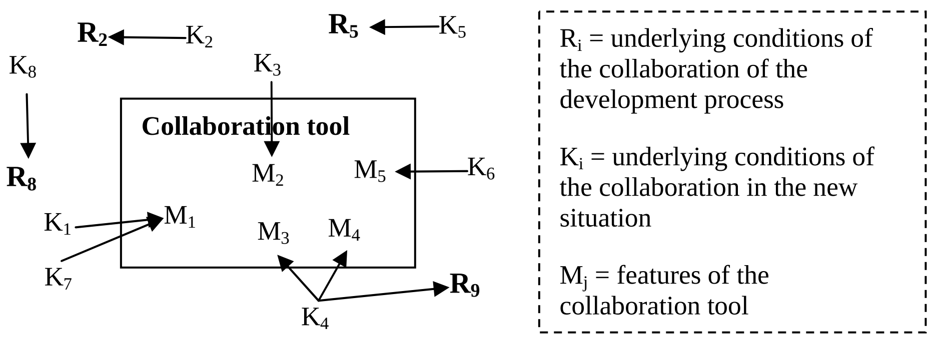

Figure 2: Use of a collaboration tool in a new environment

Simultaneously all information about the collaboration environment of the development situation which are not mapped onto a feature of the tool are lost. This problem can be counteracted by adding the relevant environmental parameters and its values at the time of development to the description of the collaboration tool. These environmental parameters can then be user as decision criteria during a selection process for the rating of alternatives (figure 2).

4 Relevant Parameters of a Collaboration Environment

Environmental parameters describing a collaboration situation are defined on the basis of the concept of collaboration. Basic elements of collaboration are the co-operation, the communications and co-ordination processes as well as the context. The context is subdivided into an organisational context, a team context and an individual context each individual collaboration partner brings along (figure 3). The environmental parameters result from the characteristics the co-operation and co-ordination, the communication processes and the different contexts are specified by.

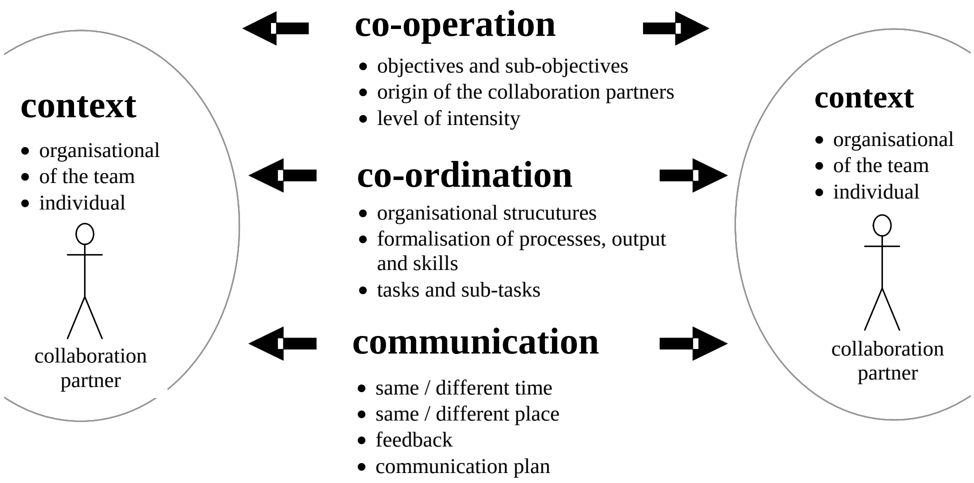

Figure 3: concept of collaboration and its environmental parameters – following [Ra05]

The resultant collaboration environment is complex. The value of the different parameters in the sub-processes the software development process is composed of varies as there are different individual collaboration partners and teams involved in the sub processes. Subject to the sub-process one or two environmental parameters become more important for the collaboration tool. For example, if the collaboration tool supports the process of decision-making the decision-making structures and competences, being part of the parameters of co-ordination, will be relevant. Furthermore the objectives set by the co-operation agreement which belong to the parameters of co-operation are important as well. The existence of a communication strategy or the definition of a communication plan being part of the parameters of communication, for example, is of minor importance for a sub-process of decision making.

By knowing the values of the environmental parameters the equivalence of collaboration situations can be assessed. Even so the parameters do not comprise information specific to an organisation or a project. *'Objectives'* for example does not describe the concrete contents of the co-operation agreement but provides information whether strategic objectives are agreed upon, whether these strategic objectives are subdivided into operative sub-objectives, whether every collaboration partner has agreed upon the designated objectives and if it is known or suspected one or two of the collaboration partners have their own agenda.

5 Environmental Parameters of OpenProsal

OpenProposal [RB07] is a collaboration tool supporting the requirement engineering process. The fundamental idea behind OpenProposal stems from the fact that in most modern software products the user's requirements refer directly to the graphical user interface. It therefore stands to reason to capture these requirements in a graphical form, supplementing a textual description. OpenProposal is supposed to allow users to annotate their feature requests, error reports or enhancement requests directly on their applications workspace and send these requests to requirements management. A lot of communication problems can thus be avoided – e.g. misconceptions due to wrong choice of wording, incomplete data, descriptions which are too elaborate – which often arise from text-only communication like E-mail or an internal employee suggestion systems. The aim of OpenProposal is to integrate users efficiently into the development process during their daily routine when using the application, to reduce the usual effort associated with participative requirements elicitation and to allow a high degree of implementation of the captured requirements through the structured recording. Furthermore, OpenProposal is supposed to increase the transparency of the requirements for the management process, thus ensuring motivated participation of as many employees as possible during requirements elicitation.

OpenProposal is available as a prototype. So far two evaluation processes have been carried out. The first evaluation process comprised usability tests and took place with 15 test persons (computer science students and scientists at the University of Karlsruhe). Every person was interviewed separately. The interviews lasted approximately 90 minutes. The test persons had to solve problems with OpenProposal and to asses the usability of the prototype and to review the concept of the OpenProposal approach. The second evaluation process took place at TRUMPF, a high-tech company focusing on production and medical technology. The TRUMPF Group is one of the world's leading companies in manufacturing technology, with sales of € 1.65 billion/US$ 2 billion and approximately 6500 employees. OpenProposal was tested in several usability workshops and became now an inherent part of the usability process at TRUMPF. In summary, the following statements concerning the usability of the supposed advantages of and raised fears related to OpenProposal were made.

Positively noticed was that OpenProposal

- helps to avoid misunderstandings
- reduces the complexity of the communication processes

- is comprehensible and easily learned
- enables the users to trace the status of the requirements.

Reservations noted were that
- the total job time will take longer with the use of OpenProposal,
- unneeded or redundant proposals are anticipated and
- OpenProposal is competitive to other existent collaboration tools.

In general the test persons presented very varying problem solving strategies and the user's advantages being a consequence of the flexibility and the licences of the tool resulted in disadvantages for the developer. Part 5.1 to part 5.5 demonstrates the deduction of presumptions concerning the acceptance of OpenProposal from the statements made by the test persons. Environmental parameters of the collaboration are defined and analysed with regard to the consequences of the use and success of OpenProposal.

5.1 Environmental Parameter: origin of the collaboration partners

Misunderstandings in the requirement engineering process occurs if the involved collaboration partners come from diverging environments which is reflected in different views on and knowledge about the facts and circumstances of the business processes. Knowledge about the *'origin of the collaboration partners'* is important for the analysis of the alleged acceptance factor that OpenProposal helps to avoid misunderstandings.

Different organisational structures in which the collaboration partners are working have to be considered according to their membership to a professional group. Professional groups participating in the software development process are the group of the users, the group of the software developers and the group of external consultants [BR05]. Eventually there are problems of comprehension as they are 'speaking a different language' (they do not use the same vocabulary). Working together with external consultants can lead to problems concerning decision making responsibilities, competencies and authorities to instruct. For example, if the collaboration partners come from different organisational environments with different underlying structures misunderstandings can result from the understanding of self-responsible working or of the significance of independent decision-making. The weight of the parameter can be balanced by the values of the parameter *'individual context'* of every single collaboration partner if they indicate a similar individual background regarding their skills.

OpenProposal helps to avoid misunderstandings by using electronic documents bearing the same meaning for all involved parties as a basis. The document takes over the role of the description of the starting position. OpenProposal makes available a set of functionality for annotating the electronic resource. The implication of the functionality is unambiguous. Primordially OpenProposal was developed for the co-operation between users and software developers. If all the collaboration partners belong to the professional group of software developers it can be supposed that no misunderstandings are to be expected. So the originally positively noted fact is not a real advantage of OpenProposal.

The alleged acceptance factor can change to the contrary into a disadvantage of OpenProposal. The use of OpenProposal would mean an additional step of work to the software developer as he has to transform the requirement recorded by OpenProposal into a notation suitable for the further development process. If a software developer has to elicit and describe requirements he would record them directly in this notation.

The *‘origin of the collaboration partners’* is only known as an underlying condition of the collaboration environment of the development and cannot be represented in a feature of a collaboration tool. So if the manual of the collaboration tool does not contain any information about the value of this parameter this information will be lost by transferring the tool into a different collaboration environment.

5.2 Environmental parameter: feedback

To inform the user about the status of the requirements recorded by him means feedback. Positive feedback is especially motivating and encourages the user to participate further in the process of software development. Providing the possibility of feedback by OpenProposal contributes to the acceptance of the collaboration tool as the user receives feedback about the value of his participation. However feedback is only a factor increasing acceptance if it is part of the *‘information culture’* of the company. If feedback is not installed as a *‘motivation factor’* in the company the collaboration partner is not used to deal with feedback. Potentially negative feedback is feared so the alleged acceptance factor can change to the contrary into a disadvantage of OpenProposal as well.

The parameter *‘feedback’* is represented by a functional requirement of the collaboration tool and can be mapped onto a feature of the tool.

5.3 Environmental parameter: level of intensity

The *‚level of intensity of the collaboration’* is used in the research area of co-operation. It describes the profundity of the exchange of information between the collaboration partners and if the information is developed on the division of labour or in intense common action [Fo94, Kl95, TL04, ZSM03]. Values of the parameter *‘level of intensity of the collaboration’* are ‘exchange of information’, ‘co-ordinated action based on the division of labour’ and ‘common action’. The level of detail of the information is increasing from the value ‘exchange of information’ to ‘common action’. The higher the degree of information the closer is the co-operation. ‘Exchange of information’ means that the collaboration partners inform each other about the activities carried out for achieving the common objectives. The co-operation is more intensive if ‘action based on the division of labour’ is agreed upon. The activities are split up and the results of the work will be communicated. ‘Common action’ is the highest level of intensity. The results are developed in close contact.

OpenProposal was developed for the *‘level of intensity’* of ‘exchange of information’. If this is true for the current collaboration situation no assessment about a probable additional job time can be made. This depends on different parameters. But if the current collaboration situation is planed for the *'level of intensity'* of 'common action' between users and software developers for the elicitation of requirements the fear of longer job time will become true. The use of OpenProposal means an additional working step as the software developer has to transform the requirements into a notation suitable for the development process afterwards (cf. 5.1). (But this is not the same case as described in 5.1 as in 5.1 a software developer records the requirements and sends them to a another software developer. Here a software developer and a user record the requirements in common work and send them to (another) software developer). The additional job time has a negative impact on the acceptance of the collaboration tool.

As the *'origin of the collaboration partners'* the *'level of intensity'* is only known as an underlying condition of the collaboration environment of the development the information about the value of this parameter will be lost by transferring the tool into a different collaboration environment as well.

5.4 Environmental parameter: technology structure

The '*technology structure*' has to be rated regarding the current state and the further development. The interaction between the collaboration tool and the current technology structure in a project [Ma00] and the possibilities of integrating the tool into the working procedures [RB06] are elementary for the acceptance of a collaboration tool. Not obtaining sufficient support is obstructive to the process of the acceptance of a tool. Particularly if the collaboration tool only supports a small sub process of the software development process the including of the tool into the *'technology structure'* is essential.

Reservations noted that OpenProposal is competitive to other existent collaboration tools – a high trust in the ability of the technicians to create an interface to other existent tools based on a good technology structure had a positive influence on the acceptance of OpenProposal. During the second evaluation phase of OpenProposal the users presented autonomous proposals for the integration of the new tool into their working procedure. It was therefore considered as a productive completion to existent tools.

The requirements resulting from the complexity of the integration of the collaboration tool into the infrastructure of the collaboration environment are mapped onto non-functional requirements resulting in features of the tool. The volume of these requirements combined with the maturity of the *'technology structure'* of the new collaboration environment result in trust in the ability of the technicians to integrate the tool into the infrastructure. The existence of this trust is a precondition for the acceptance of a tool.

5.5 Environmental parameter: individual context

The individual context of each collaboration partner has a significant influence on the acceptance and adoption of a collaboration tool in a company or a team. Technical skills, personal competency to be open to new ideas and a general common basis for co-operation are preconditions for collaboration [MP01, OO00, Re05]. The ‘individual context’ comprises among others *‘technical skills’* and *‘cognitive skills’* [EW04, We04, Sp96, Ko02].

OpenProposal is comprehensible and easily learned so the adoption of OpenProposal does not require above-average *‘technical skills’*. In general the test persons presented very varying problem solving strategies. All these strategies could be supported by OpenProposal. The use of OpenProposal requires *‘cognitive skills’* in so far as the user should be able to describe a problem formally.

During the development of the collaboration tool the users’ skills influenced the development process tacitly. They represent underlying conditions which cannot be mapped onto features. The context of the original collaboration partners will be lost by transferring the tool into a new collaboration environment.

5.6 Overview of the environmental parameters of OpenProposal

The listing of advantages of a collaboration tool is not equivalent to the success and acceptance of the tool as the analysis of the results of the evaluation phases of OpenProposal manifests. The acceptance of the tool depends rather on the concrete collaboration environment a collaboration tool is intended to be used in. The specific configuration of this environment is responsible for the alleged advantages to the tools being real advantages for the users. Only in that case there will be acceptance for the adoption of the tool. Table 5 outlines the deduced environmental parameters of OpenProposal.

Environmental parameters	**relation to OpenProposal**
origin of the collaboration partners	helps to avoid misunderstandings (cf. 5.1)
level of intensity	fear of additional job time (cf. 5.3)
technology structure	definition of interfaces to existent tools (cf. 5.4)
feedback	tracing of the status of requirements (cf. 5.2)
information culture	handling of feedback (cf. 5.2)
motivation system	feedback as a part of the motivation system (cf. 5.2)
individual context	OpenProposal is comprehensible and easily learned. The technical and cognitive skills of the users vary significantly (cf. 5.5)

Table 1: environmental parameters of OpenProposal

The technical and cognitive skills of the users varied significantly. So OpenProposal was used in very different modes. The less technical user first of all used the functionality of comment and considered the storage and sending of the automatic screen-shot being the utmost advantage of OpenProposal. The more technical user exploited all the functionalities of OpenProposal and submitted proposals for additional functionality. That led to reflections about product variants of OpenProposal. One of the variant will have a reduced set of functionalities and is designated for users with lower technical and cognitive skills. The other variant will have an expanded set of functionalities for a power-user. The primary intention of Open Proposal – helping to avoid misunderstandings - will then fade into the background. The environmental parameter *‘origin of the collaboration partners’* will change for this second product variant.

Divergences in the *‘origin of the collaboration partners’* between the collaboration environments of the development and the second evaluation situation led to the proposals for modifications of the collaboration tool. The realisation of these proposals resulted in a modification of the environmental parameters so that the divergences were reduced. Alternatively product variants were created so the collaboration tool matched different collaboration environments. Obviously the development of optimal product variants of collaboration tools respectively functionalities optimally meeting the needs of users can be supported by the analysis of collaboration environments.

6 Summary and future prospects

The reflection of the processes of development and evaluation of OpenProposal under the view of the collaboration environment led to a definition of relevant environmental parameters for the process of ‘Elicitation of problem and change requests’. Different conclusions can be drawn from these environmental parameters. Firstly due to divergences in the values of the environmental parameters of the development and the operational environment statements concerning expected problems of acceptance of the collaboration tool by the users can be made. The peculiar characteristics of collaboration tools can be taken into account by integrating these environmental parameters into the selection procedure of collaboration tools. Secondly functionalities optimally meeting the needs of users respectively the development of optimal product variants of collaboration tools can be supported by the analysis of the collaboration environments. Thus an adaptation of the tool to the designated collaboration environment can take place which results in higher acceptance of the tool.

Further research will firstly comprise the definition of environmental parameters permitting to describe a collaboration situation broadly. This definition will be based on the concept of collaboration (cf. chap. 4). These environmental parameters can then be taken as a supplement to existent selection procedures for software with the aim to adapt the procedure to the selection of collaboration tools. The adoption of these parameters will require the identification of the relevant environmental parameters for the process which has to be supported by the collaboration tool. The relevant parameters have to meet the peculiar characteristics of the processes. Only these parameters should then be used for rating the collaboration tools coming into question. Secondly the selection of the relevant environmental parameters will be realised for selected sub-processes of the software development process and considerations in regard to integrate these environmental parameters into process models will be made.

References

[Al99] Altmann, J.: Kooperative Softwareentwicklung. Universitätsverlag Rudolf Trauner, 1999.

[An03] Andriessen, J.H.E.: Working with Groupware. Berlin, Heidelberg, Springer Verlag, 2003.

[BH04] Bardram, J.E.; Hansen, T.R.: The AWARE Architecture: Supporting Context-Mediated Social Awareness in Mobile Cooperation. In: Proceedings of the CSCW '04, Chicago, Illinois, USA, pp.192-201, 2004.

[BOO95] Bekker, M.M; Olson, J.S.; Olson, G.M.: Analysis of gestures in face-to-face design teams provides guidance for how to use groupware in design. In: Proceedings of the DIS '95, Ann Arbor, USA, pp.157-166, 1995.

[BR05] Behm, A.; Rashid A.: Kollaborationspunkte im Softwareentwicklungsprozess. Working Paper of the ‚FZI Forschungszentrum Informatik an der Univ. Karlsruhe', 2005.

[BS00] Borghoff, U.M.; Schlichter, J.H.: Computer-Supported Cooperative Work – Introduction to Distributed Applications. Springer Verlag, Berlin, Heidelberg, New York, 2000.

[Bu07a] Bundesamt für Sicherheit in der Informationstechnik: B 1.10 Standardsoftware. http://www.bsi.bund.de/gshb/deutsch/baust/b01010.htm, [July, 11, 2007].

[Bu07b] Bundesamt für Sicherheit in der Informationstechnik: M 2.80 Erstellung eines Anforderungskatalogs für Standardsoftware. http://www.bsi.bund.de/gshb/deutsch/m/m02080.htm, [July, 11, 2007].

[CFQ07] Carvallo, J.P.; Franch, X.; Quer, C.: Determining Criteria for Selection Software Components: Lessons Learned. In: IEEE Software, pp.84-94, May/June 2007.

[CNH04] Convertino, G.; Neale, D.C.; Hobby, L.; Carroll, J.M.; Rosson M.B.: A Laboratori Method for Studying Activity Awareness. In: The Proceedings of the NordiCHI '04, Tampere, Finnland, pp.313-322, 2004.

[Co04] CollaBaWü, Forschungsantrag (proposal of research): Kollaborative, komponentenbasierte Entwicklung von Unternehmenssoftware im Finanzdienstleistungsbereich von Baden-Württemberg, Forschungsverbund ‚Unternehmenssoftware', Land Baden-Württemberg, www.collabawue.de [November, 26, 2007], 2004.

[Co05] Collaboration Strategies. *Inside Collaboration Newsletter*, 4, 1 (2005). Available: http://www.collaborate.com/publication/newsletter2/nl0105.html, [November, 05, 2007].

[CO07] Cherian, S.P.; Olson, J.S.: Extending a Theory of Remote Scientific Collaboration to Corporate Contexts. In: Proceedings of the CHI '07, San Jose, USA, pp.2321-2326, 2007.

[DL86] Daft, R.L.; Lengel, R.H.: Organizational Information Requirements, Media Richness and Structural Design. In: Management Science, vol.32, no.5, May 1986.

[Eh99] Ehrlich, K.: Designing Groupware Applications: A Work-Centered Design Approach. In: Beaudouin-Lafon: Computer Supported CooperativeWork, JohnWiley & Sons Ltd, 1999.

[EGR91] Ellis, C. A.; Gibbs, S. J.; Rein, G. L.: Groupware: Some issues and experiences. In: Communications of the ACM, 34, 1, pp. 39-58, 1991.

[EW04] Edwards, A.; Wilson, J. R.: Implementing Virtual Teams, Gower Publishing Company, 2004.

[FLP06] Frank, E.; Liesebach, K.; Pfitzmann K.B.: Privacy-Aware User Interfaces within Collaborative Environments. In: Proceedings of the international workshop in conjunction with AVI 2006 on CAI'06, Venice, Italy, pp.45-58, 2006.

[Fo94] Fontanari, M.: Kooperationsgestaltungsprozess in Theorie und Praxis. Duncker&Humblot, Berlin, 1994.

[Gr01] Gronau, N.: Industrielle Standardsoftware – Auswahl und Einführung. Oldenbourg, Wissenschaftsverlag GmbH, 2001.

[Gr94] Grudin, J.: Groupware and social dynamics: Eight challenges for developers. In: Commununications of the ACM 37, 1, pp.92–105, 1994.

[GPS04] Gutwin, C.; Penner, R.; Schneider, K.: Group Awareness in Distributed Software Development. In: Proceedings of the CSCW ’04, Chicago, Illinois, USA, pp.72-81, 2004.
[HI05] Hawkey, K.; Inkpen, K.M.: Privacy Gradients: Exploring Ways to Manage Incidental Information During Co-located Collaboration. In: CHI '05 extended abstracts on Human factors in computing systems, Portland, Oregon, USA, pp.1431-1434, 2005.
[HL01] Hofmann, H.F.; Lehner, F.: Requirements: Engineering as a Success Factor in Software Projects. In: IEEE Software, vol. 19, no. 3, pp. 58-66, 2001.
[HM03] ter Hofte, G.H.; Mulder, I.; Dynamic Personal Social Networks: a new perspective for CSCW research and design. In: ACM SIGGROUP Bulletin, vol. 24, no.3, pp.139-142, 2003.
[JMS02] Juristo, N.; Moreno, A.M., Silva, A.: Is the European Industry Moving toward Solving Requirements Engineering Problems? In: IEEE Software, vol. 19, no. 6, pp. 70-77, 2002.
[Kl95] Klanke, B.: Kooperationen als Instrument der strategischen Unternehmensführung. Analyse und Gestaltung – dargestellt am Beispiel von Kooperationen zwischen Wettbewerbern. Dissertation, University of Münster, 1995.
[KL99] Kristoffersen, S.; Ljungberg, F.: An Empirical study of How People Establish Interaction. Implications for CSCW Session Management Models. In: Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit CHI '99, Pittsburgh, USA, 1999.
[Ko02] Kotulla, A.: Management von Softwareprojekten – Erfolgs- und Mißerfolgsfaktoren bei international verteilter Entwicklung. Deutscher Universitätsverlag GmbH, 2002.
[Ma00] Majchrzak, A.; Rice, R.E.; Malhotra, A.; King, N.; Ba, S.: Technology Adaptation: The Case of a Computer-supported Interorganizational Virtual Team. In: MIS Quarterly, vol.24, no.4, pp.569-600, Dec.2000.
[MC94] Malone, T. W.; Crowston, K.: The Interdisciplinary Study of Coordination. In: ACM Computing Surveys, 26, 1, pp.87-119, 1994.
[MP01] Mark, G.; Poltrock, S.: Diffusion of a Collaborative Technology Across Distance. In: Proceedings of the GROUP ’01, Boulder, Colorado, USA, pp.232-241, 2001.
[OO00] Olson, G.; Olson, J.: Distance matters. In: Human Computer Interaction, vol. 15, pp. 139-179, 2000.
[Pa01] Pankoke-Babatz U.; Prinz, W.; Wulf, V.; Rohde M.: Entwicklung von Groupware als Balance zwischen Anwendern und technischer Innovation. In: Schwabe, Gerhard; Streitz, Norbert; Unland, Rainer (Hrsg.): CSCW Kompendium - Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten. Berlin: Springer, pp. 373-394, 2001.
[Ra05] Rashid, A.; Behm, A.; Geisser, M.; Hildenbrand, T.: Kollaborative Softwareentwicklung – Zum Kollaborationsbegriff. Working Paper of the ‚FZI Forschungszentrum Informatik an der Univ. Karlsruhe’, 2005.
[RB06] Rama, J.; Bishop, J.: A Survey and Comparison of CSCW Groupware Applications. In: Proceedings of SAICSIT 2006, pp. 198-205, 2006.
[RB07] Rashid, A.; Baumann, J.: OpenProposal: Visual Requirement Specification In End-User Participation. In: Proceedings of the 15th IEEE International Requirements Engineering Conference, October 15-19th, New Delhi (India), 2007.
[Re05] Redmiles, D.; Wilensky, H.; Kosaka, K.; de Paula, R.: What Ideal End Users Teach Us About Collaborative Software. In: Proceedings of the GROUP ’05, Sanibel Island, USA, pp. 260-263, 2005.
[RI06] Razavi, M.N.; Iverson, L.: A Grounded Theory of Information Sharing Behavior in a Personal Learning Space. In: Proceedings of the CSCW ’06, Banff, Alberta, Canada, pp.459-468, 2006.
[SH05] Stahlknecht, P.; Hasenkamp, U.: Einführung in die Wirtschaftsinformatik, 11. Auflage, Springer Verlag, 2005.

[Sp96] Spieß, E.: Kooperatives Handeln in Organisationen. Theoriestränge und empirische Studien. Rainer Hampp Verlag, München und Mering, 1998.

[Te00] Teasley, S.; Covi, L.; Krishnan, M.S.; Olson, J.S.: How Does Radical Collocation Help a Team Succeed? In: Proceedings of the CSCW ’00, Philadelphia, USA, 2000.

[Te02] Tewoldeberhan, T.W.; Verbraeck, A.; Valentin, E.; Bardonnet, G.: An Evaluation and Selection Methodology for discrete-event Simulation Software. In: Proceedings of the 2002 Winter Simulation Conference, 2002.

[TGG06] Tee, K.; Greenberg, S.; Gutwin, C.: Providing Artifact Awareness to a Distributed Group through Screen Sharing. In: Proceedings of the CSCW ’06, Banff, Alberta, Canada, pp.99-108, 2006.

[Th07] Thissen, M.R.; Page, J.M.; Bharathi, M.C.; Austin, R.L.: Communication Tools for Distributed Software Development Teams. In: Proceedings of the SIGMIS-CPR ’07, St. Louis, Missouri, USA, 2007.

[TL04] Theling, T.; Loos, P.: Determinanten und Formen von Unternehmenskooperationen. In: Working Papers of the Research Group Information Systems & Management, paper 18, ISSN 1617-6332 (Internet version), 2004.

[We04] Weinert, A.B.: Organisations- und Personalpsychologie, Lehrbuch, 5. Auflage. Beltz Verlag Weinheim Basel, 2004.

[We06] Weiseth, P.E.; Munkvold, B.E.; Tvedte, B.; Larsen, S.: The Wheel of Collaboration Tools: A Typology for Analysis within a Holistic Framework. In: Proceedings of the CSCW ’06, Banff, Alberta, Canada, 2006.

[ZB98] Zigurs, I.; Buckland, B.: A Theory of Task/Technology Fit and Group Support Systems Effectiveness. MIS Quarterly, 22, 3, pp.313-334, 1998.

[ZSM03] Zentes, J.; Swoboda, B.; Morschett, D.: Perspektiven der Führung kooperativer Systeme. In: Zentes, J.; Swoboda, B.; Morschett D. (Hrsg.): Kooperationen, Allianzen und Netzwerke, Grundlagen – Ansätze – Perspektiven. Gabler Verlag, 2003.