Herskovic, V., J. Pino, S. Ochoa and P. Antunes (2007) Evaluation Methods for Groupware Systems. Groupware: Design, Implementation, and Use. 13th International Workshop, CRIWG 2007, Bariloche, Argentina, September 2007 Proceedings. J. Haake, S. Ochoa and A. Cechich. Lecture Notes in Computer Science, vol. 4715, pp. 328-336. Heidelberg, Springer-Verlag.

Evaluation Methods for Groupware Systems

Valeria Herskovic[1], José A. Pino[1], Sergio F. Ochoa[1], Pedro Antunes[2]

[1] Department of Computer Science, Universidad de Chile, Chile
{vherskov, jpino, sochoa}@dcc.uchile.cl
[2] Department of Informatics, University of Lisbon, Portugal
paa@di.fc.ul.pt

Abstract. Collaborative systems evaluation is necessary in several situations. However, evaluation is frequently done in an ad-hoc manner or not at all. This paper presents a survey of evaluation methods for groupware systems. The analysis, comparison and classification of these methods will help developers choose the appropriate methods for their situation. Furthermore, the survey allows identification of strengths and weaknesses of existing methods, opening opportunities for research in this area. The proposed comparison criteria represent a framework to evaluate and classify new evaluation methods.

1 Introduction

The evaluation of collaborative systems is an important yet not fully solved problem in the field of Computer Supported Cooperative Work (CSCW). Despite the need for evaluation, many groupware systems are deficiently evaluated. A study of 45 articles from 8 years of the CSCW conference revealed that almost one third of the presented groupware systems were not evaluated in a formal way [12], while a study that also included the ECSCW conference and the Journal of CSCW found few articles that focused on evaluation [15]. Even when evaluations are done, many of them are ad-hoc, depending on researchers' interests or appropriateness for a specific setting [11].

The possible reasons for the lack of widespread groupware evaluation are various. First, methods for single-user systems are not always applicable to groupware, since the outcome depends on the various backgrounds of group members, organizational culture and group dynamics [5]. Second, evaluation may be expensive and the required resources may be unavailable [2]. Third, a groupware system's benefits may be long-term, so group observation should extend over long periods [5]. Finally, it is not easy for an evaluator to identify which methods to apply in a particular situation.

A survey of the evaluation methods that are currently available for groupware systems is presented. Only strategies explicitly designed for, or adapted to groupware systems are considered, since these products have particular characteristics that may render other evaluation methods inapplicable. The paper is organized as follows. Section 2 briefly describes each of the reviewed methods. Section 3 presents their comparison and categorization. It also describes a general strategy to choose the most appropriate evaluation method. Section 4 proposes a strategy to minimize the cost to evaluate a groupware system. Section 5 presents the conclusions and further work.

2 Analysis of the Methods

This section presents a summary of the groupware evaluation methods. Each one is briefly explained to give an overview of it and the steps that must be followed to apply it. These methods can be directly used to evaluate a groupware application, or they can be part of a macro evaluation strategy. In the latter case, the global strategy may ask for iterative evaluation [8], or that it should consider several stakeholders [16]. The evaluation methods in this survey are those directly applicable to groupware systems. They are the following ones:

Groupware Heuristic Evaluation (GHE). GHE is an adaptation of the Heuristic Evaluation method, in which single-user systems are evaluated by visually inspecting the interface and judging its compliance with usability principles. GHE is based on eight groupware heuristics [4], which act as a checklist of characteristics a collaborative system should have. Evaluators who are experts in them examine the interface, recording each problem they encounter, the violated heuristic, a severity rating and optionally, a solution to the problem. The problems are then filtered, classified and consolidated into a list, which is used to improve the application.

Groupware Walkthrough (GWA). GWA is an evaluation method based on cognitive walkthrough, an inspection technique for single-user software [13]. In GWA, a scenario is a description of an activity or set of tasks, which includes the users, their knowledge, the intended outcome, and circumstances surrounding it. In order to construct scenarios, evaluators observe users and identify episodes of collaboration. Each evaluator, taking the role of all users or one in particular, walks through the tasks in a laboratory setting, recording each problem he encounters. A meeting is then conducted to analyze the results of the evaluation.

Collaboration Usability Analysis (CUA). CUA is a task analysis technique focused on the teamwork aspects of collaboration in shared tasks [14]. It provides high-level and low-level representations of the collaborative situation and task to be studied, and multiple ways to represent actors and their interactions. CUA proposes that each collaborative action can be mapped to a set of collaboration mechanisms, or fine grain representations of basic collaborative actions, which may be related to elements in the user interface. The resulting diagrams capture details about task components, a notion of the flow through them and the task distribution.

Groupware Observational User Testing (GOT). GOT is a technique based on the observational user testing method (OUT). OUT involves evaluators observing how users perform particular tasks supported by a system in a laboratory setting [6]. Evaluators either monitor users having problems with a task, or ask users to think aloud about what they are doing to gain insight on their work. GOT follows the same principle, but focuses on collaboration and analyzes users' work through predefined criteria, e.g., the mechanics of collaboration [6].

Human-Performance Models (HPM). HPM describe how a person interacts with a physical interface at a low level of detail based on a cognitive architecture, e.g., the keystroke level model (KLM) approximates the interaction of a single user with an interface. HPM adapts this model to a group of users communicating through a shared workspace [2]. In this method, evaluators first decompose the physical interface into several shared workspaces. Then, they define critical scenarios focused on the

collaborative actions for the shared workspaces. Finally, evaluators compare group performance in the critical scenarios, e.g., using KLM to predict execution times.

"Quick-and-dirty" Ethnography (QDE). Ethnography refers to the qualitative description of human social phenomena to produce detailed descriptions of the work activities of actors within specific contexts. QDE [9] aims to adapt ethnography to evaluation. Here, evaluators do brief workplace studies to provide a general sense of the setting for designers. QDE accepts the impossibility of gathering a complete understanding of the setting, providing a broad understanding instead. It suggests the deficiencies of a system, supplying designers with the key issues that bear on acceptability and usability, thus allowing existing and future systems to be improved.

Performance Analysis (PAN). PAN is an evaluation method that allows formal analysis of a groupware application [3]. The application to be studied is modeled as a task to be performed by a number of people in a number of stages, and the concepts of result quality, time, and total amount of work done are defined. The evaluators must define a way to compute the quality (e.g., group recall in a collaborative retrieval task), and maximize the quality vs. work done either analytically or experimentally.

Perceived Value (PVA). PVA measures the perceived value (PV) attributed to a meetingware system by its users [1]. This method tries to measure the organizational impact and the alignment between system capabilities and developers' and users' expectations. Developers begin by identifying relevant components for system evaluation. Then, users and developers negotiate the relevant system attributes to be evaluated by users. After the users have worked with the system, they fill out an evaluation map by noting whether the components support the attributes or not. Using these ratings, a metric that represents the PV is calculated.

Scenario-Based Evaluation (SBE). SBE provides evaluators with realistic settings in which to base their evaluations. A scenario is a detailed description of an activity, which includes the task, actor, context and claims, which are statements about using the system. In a field evaluation using SBE [7], evaluators perform semi-structured interviews of the users to discover scenarios and claims about them. Then, focus groups validate these findings. The frequency and percentage of positive claims help quantify the organizational contributions of the system, and the positive and negative claims about existing and envisioned features provide information to aid in redesign.

Cooperation Scenarios (COS). The COS method aims to capture users' work and its context [17]. Scenarios (SC) are descriptions of work practices, including motivation and goals. In order to construct SC, evaluators conduct field studies, semi-structured interviews, and workplace visits. Through these activities, they identify cooperative behavior, users involved in it, their roles and the relevant context. For each role involved in the cooperative activity, evaluators analyze the new design to see how the task changes and who benefits from the new technology. Then, the prototype is presented as a SC in a workshop with users to discover design flaws.

E-MAGINE (EMA). EMA is a method based on two concepts: (1) a system should match its environment, and (2) the perception of the user is important [10]. EMA begins with a meeting between client and evaluator, in which the goals are set. Then, a quick semi-structured interview with someone familiar with the group is applied to build a profile of the group and scenario. It also guides the selection of evaluation

tools, and the issues that will be evaluated, such as social cohesion and usability. Finally, the results are fed back to the group to apply the proposed changes.

Knowledge Management Approach (KMA). This method posits that knowledge is the most important asset of organizations. Evaluation using KMA measures whether the system helps users detect knowledge flows and disseminate, store and reuse knowledge [19]. The knowledge circulation process is comprised of six phases (knowledge creation, accumulation, sharing, utilization, internalization), which are also the areas to be evaluated by this approach. To perform evaluation, each area has a list of associated questions, which may be used as a checklist by evaluators.

3 Selecting Groupware Evaluation Methods

Whenever a stakeholder needs to choose a groupware evaluation method, she does it for a specific context. For instance, a project manager may want to determine how well the functionality of a groupware system under development matches the expectations of an organization. Such context allows the manager to consider some key features of each method in order to establish which ones could be appropriate.

Considering several features in the selection process will make the list of potential methods short and accurate. If the suitable methods list is empty, then an ad-hoc evaluation method should be designed. If the list contains more than one method, then the evaluator can choose one based on a prioritization of their key features. We call these key features of a method its *classification criteria*. The next sections present three classifications of groupware evaluation methods.

3.1 Classification based on stakeholders and product state

Table 1 presents a classification that considers the concerned stakeholders (developers, users and the organization) and the state of the product (under development or finished). A brief explanation of each category is included.

Evaluation methods for systems under development. While a collaborative system is under construction, the *developers* require formative and inexpensive evaluation methods that allow them to test the product, discover its flaws, and redesign it accordingly. These methods are usually done in a laboratory setting without users. On the other hand, *users* of this system could be interested in ensuring that the system works as desired and allows effective and efficient collaboration. Evaluation methods must thus involve users and focus on their opinions. Finally, the *organization* as a whole requires that a collaborative system improves work, efficiency and the quality of results, allowing managers to justify investments in the technology.

Evaluation methods for finished products. Organizations acquiring a groupware system may require *developers* to adapt the product to their needs. Also, the development team may need to conduct a post-mortem analysis of a finished product. Evaluation methods for developers must thus measure the matching between product functionality and the users and organizational needs. These methods must be summative and supply information to help developers to improve the system. On the other hand, the *users* of the system need to ensure the system works as desired and allows effective and efficient collaboration. Similar to the previous case, methods

must involve users and focus on their opinions. However, users now have the finished product to experiment with, so methods may be summative. The final case is when an *organization* acquires a groupware system and it must go through the adoption of the technology. In this case, evaluation methods should be summative, tested in the organization's environment, and measure how well a system fits in the organization.

Table 1: Method categorization based on stakeholders and product state

	Developers	Users	Organization
Products Under Development	GHE, GWA CUA, HPM, PAN	SBE, COS EMA, KMA	
Finished Products		GOT, QDE	GOT, QDE PAN, PVA

It is possible to identify an initial set of relevant methods for a particular scenario considering these criteria. Stakeholders' identification and product state can be used as initial evaluation criteria because they are fast to instantiate and highly relevant.

3.2 Classification based on type, scope and duration

Table 2 classifies the evaluation methods considering the people's participation, time of application, evaluation type, place, time span and goal. This table should be used in the same way as table 1 in order to perform the selection process. The classification criteria included in table 2 are briefly described below.

People participation states who participates in the evaluation besides evaluators, usually users (U), developers (D), experts (E), or combinations of them. This criterion helps determine the viability of a method based on human resources availability.

Table 2: Classification of evaluation strategies

Evaluation Method	Who			When			Type		Loc.		Time Span			Goal		
	U	D	E	B	S	F	N	Q	W	L	H	Y	K	P	C	X
Groupware Heuristic Evaluation (GHE)			X		X	X		X		X	X			X		
Groupware Walkthrough (GWA)					X	X		X		X		X			X	
Collaboration Usability Analysis (CUA)					X	X		X		X		X			X	
Groupware Observ. User Testing (GOT)	X				X	X		X		X			X		X	
Human-Performance Models (HPM)				X	X	X	X			X		X			X	
Quick-and-Dirty Ethnography (QDE)	X				X			X	X				X			X
Performance Analysis (PAN)					X	X	X			X		X		X		
Perceived Value (PVA)	X	X		X	X	X		X	X			X		X		
Scenario Based Evaluation (SBE)	X				X	X		X	X				X		X	X
Cooperation Scenarios (COS)	X					X		X	X				X		X	X
E-MAGINE (EMA)	X				X	X		X	X						X	X
Knowledge Management Approach (KMA)					X	X		X	X	X		X		X		X

The *time to apply the method* may be: before the system is designed to test its feasibility (B); during the development process as a formative evaluation to identify redesign needs (D); or when the application is finished (F) as a summative evaluation. This criterion helps in the selection depending on the level of progress of the project.

The *evaluation type* establishes whether the collected data is quantitative (N) or qualitative (Q). Quantitative data is useful to compare the results of several

evaluations, while qualitative data usually consists of human judgment and may be used for the most complex situations.

The *evaluation place* determines the location where evaluation is carried out, either a laboratory setting (L) or the users' actual workplace (W). Based on place availability, it is possible to determine whether a certain method may be used or not.

The *time span* of each method goes from hours (H) to days (Y) or weeks (K). This must be considered to establish whether there is enough time to do an evaluation.

The *evaluation goal* describes the objective of the evaluation, which can be to evaluate the product functionality (P), the collaboration process supported by the system (C), or the product functionality considering the collaboration context (X).

3.3 Classification based on the evaluation cost

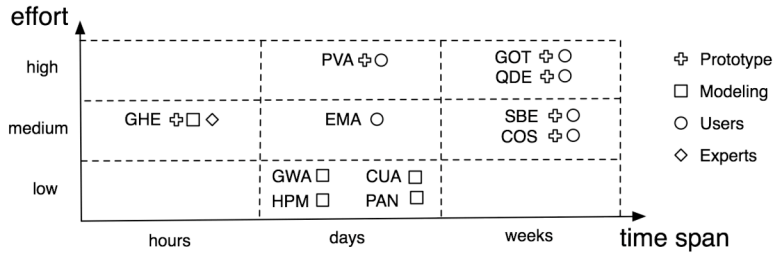

Fig. 1. Evaluation methods according to their cost

The *evaluation cost* is important during the selection of an evaluation method. A possible classification of evaluation methods is based on their cost, but this does not exist. We propose the evaluation cost be a function of the process duration and the effort required to conduct the evaluation, as shown in Fig. 1. The effort to do an evaluation was estimated based on the activities that must be done and the required human resources. If an evaluation method requires a high number of participants, then the evaluation is considered as needing a high effort. The duration of an evaluation method may be as short as a few hours, in the case of GHE, or as long as weeks for ethnographical studies. In Fig.1, methods closest to the origin are those of lowest cost, while those in the upper right corner have the highest cost. The combination of time span and effort into a single cost measure depends on the particular situation.

4 Evaluation Strategies

The previous section presented a strategy to find appropriate methods for a particular context. This section describes how to organize the evaluation process.

The high cost of evaluation is one of the reasons why groupware systems are not frequently evaluated. The evaluation process could require the use of several evaluation methods depending on how complete or accurate the diagnosis should be. An approach to evaluation should combine a first phase of purely quantitative lab-based methods with a second phase of qualitative field methods that involve the users

and their context. If necessary, it is complemented with a third phase of qualitative studies in the real work setting. This stepwise approach is derived from Twidale et al. [18], who stress the importance of context in evaluation but also believe early evaluation in an artificial environment may remove gross errors.

Each evaluation method may be categorized according to the phase in which it is optimally applied. In the first phase, major errors should be removed, while not incurring in high costs. Therefore, ideal methods for this phase are low cost, quantitative, lab-based strategies that do not require users, e.g., KMA, HPM, CUA, GWA, PAN and GHE. In the second phase, methods that require users and context can be applied. The most suitable methods for this phase are the qualitative ones, those based on user opinion, and the ones that capture the scenario of the application to test it in the lab, such as COS, EMA, and SBE. The third phase involves summative methods that should be applied in the real work setting, such as GOT and QDE.

An expensive, time-consuming approach to evaluation may be discouraging, so the costs of a three-phase evaluation must be discussed. Two aspects of applying several evaluation methods in the development of a groupware system: the cost of each one and the corresponding training cost. Focusing the most intensive evaluation efforts in the first phase should reduce the costs of subsequent evaluations, since gross errors should be discovered early on. This prevents encountering errors in the final phases of development when fixing them is most expensive. The cost of training the involved actors in each method is balanced by the various perspectives gained, which provide a comprehensive view of the application. Naturally, the number of applied methods per phase depends on how much emphasis a team wants to give evaluation, but applying one evaluation strategy per phase should provide a significant outlook into the application while not substantially increasing the costs of evaluation.

5 Conclusions and Further Work

Evaluating groupware systems is necessary and yet, many of them are not evaluated. Unevaluated systems tend to be unsuccessful because they mail fail to consider the context, stakeholders and contain errors after deployment.

The classifications proposed in this paper afford visibility to each evaluation method, allowing for fast comparison according to several criteria. The categorization also provides a tool for any interested stakeholder to choose an evaluation method that is especially useful for his particular situation. The process of choosing an appropriate evaluation method is simplified, because a short list of methods is provided according to the needs of the stakeholder as well as the product state. With this reduced list, the evaluator may choose the most appropriate method by reviewing his/her available resources (equipment, time, effort, etc) and the characteristics of each method.

The categorization of evaluation methods suggests some areas that lack appropriate evaluation methods, providing opportunities for further research, such as in the case of developers who need to modify a finished product. The comparison of the twelve methods reviewed in this paper has also highlighted the fact that most evaluation methods are qualitative, and only two of them are purely quantitative. We believe the prevalence of qualitative methods is symptomatic of the complexity of groupware, as human judgment may be required to disentangle the multiple contingencies, and ultimately appreciate if a groupware application is good or not. On the other hand, the

role of quantitative methods in CSCW evaluation is also important, since they permit the objective comparison of several applications, and may be automated. This suggests that new quantitative evaluation methods are needed. Further research in this area should improve the availability of methods for all stakeholders.

Using only one type of evaluation may prevent evaluators from gaining access to the complete picture in some cases. This suggests that several evaluation methods may be applied to obtain a comprehensive understanding of the system and its environment. CSCW systems are multifaceted. Conducting a thorough evaluation may provide additional perspective on how they function and how to improve them.

References

[1] P. Antunes, C. Costa. Perceived value: A low-cost approach to evaluate meetingware. CRIWG '03, Lecture Notes in Computer Science 2806, 109–125, 2003.

[2] P. Antunes, A. Ferreira, J. Pino. Analyzing shared workspaces design with human-performance models. CRIWG '06, Lecture Notes in Computer Science 4154, 62–77, 2006.

[3] R. Baeza-Yates, J. Pino. Towards formal evaluation of collaborative work and its application to information retrieval. Information Research, 11(4):271, 2006.

[4] K. Baker, S. Greenberg, C. Gutwin. Empirical development of a heuristic evaluation methodology for shared workspace groupware. CSCW '02, 96–105, 2002.

[5] J. Grudin. Why CSCW applications fail: problems in the design and evaluation of organization of organizational interfaces. CSCW '88, 85–93. ACM Press, 1988.

[6] C. Gutwin, S. Greenberg. The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. WETICE '00, 98–103. IEEE Comp. Soc., 2000.

[7] S. Haynes, S. Purao, A. Skattebo. Situating evaluation in scenarios of use. CSCW '04, 92–101. ACM Press, 2004.

[8] J. Huang. A conceptual framework for understanding collab. systems evaluation. WETICE '05: 14th I. Workshop on Enabling Tech., 215–220. IEEE Comp. Soc., 2005.

[9] J. Hughes, V. King, T. Rodden, H. Andersen. Moving out from the control room: Ethnography in system design. CSCW '94, 429–439. ACM Press, 1994.

[10] M. Huis in't Veld, J. Andriessen, R. Verburg. E-magine: The development of an evaluation method to assess groupware applications. WETICE '03, 153. IEEE, 2003.

[11] K. Inkpen, R. Mandryk, J. DiMicco, S. Scott. Methodology for evaluating collaboration in co-located environments. interactions, 11(6), 2004.

[12] D. Pinelle, C. Gutwin. A review of groupware evaluations. WETICE '00: 9th International Workshop on Enabling Technologies, 86–91. IEEE Computer Society, 2000.

[13] D. Pinelle, C. Gutwin. Groupware walkthrough: adding context to groupware usability evaluation. CHI '02, 455–462. ACM Press, 2002.

[14] D. Pinelle, C. Gutwin, S. Greenberg. Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration. ACM Transactions on Computer-Human Interaction, 10(4):281–311, 2003.

[15] L. Plowman, Y. Rogers, M. Ramage. What are workplace studies for? ECSCW '95: Fourth European Conference on Computer-Supported Cooperative Work, 309–324, 1995.

[16] S. Ross, M. Ramage, Y. Rogers. PETRA: Participatory evaluation through redesign and analysis. Interacting with Computers, 7(4):335–360, 1995.

[17] O. Stiemerling, A. Cremers. The use of cooperation scenarios in the design and evaluation of a CSCW system. IEEE Transact. on Software Engineering, 25:140, Jan/Feb 1999.

[18] M. Twidale, D. Randall, R. Bentley. Situated evaluation for cooperative systems. CSCW '94, 441–452, ACM Press, 1994.

[19] A. Vizcaíno, M. Martinez, G. Aranda, M. Piattini. Evaluating collaborative applications from a knowledge management approach. WETICE '05, 221–225. IEEE Comp. Soc., 2005.