WETICE 2004 ECE Workshop - Final Report

David Nutter and Cornelia Boldyreff
Faculty Of Applied Computing Sciences
University Of Lincoln,UK
LN6 7TS
{*dnutter,cboldyreff*}*@lincoln.ac.uk*

Abstract

A summary of the fifth Evaluating Collaborative Enterprises (ECE) workshop which ran on June 14th at University of Modena, Italy.

1 Introduction

The previous workshop, summarised by Raybourn et al[5] also focussed on the position of evaluation within the software lifecycle but addressed the question of evaluation design in depth instead of this workshop's focus of applied evaluation. Several papers discussed potential techniques for evaluating collaborative systems such as groupware and distributed project teams. Additionally, two papers studied the formation of virtual enterprises, including the support offered by agents and models for individual contribution. Finally, the evaluation of awareness support was also discussed in a paper.

Two issues outstanding from the 2003 workshop are relevant here. Firstly

> "In the design cycle of collaborative development, when are particular evaluation approaches effective and when are they not? Can a spectrum be developed?"

is addressed by the many papers in this year's workshop discussing the role of evaluation within a wider software development process, whether as a success indicator or a method of improving software quality and applicability. Secondly,

> "Which evaluation methods and techniques address collaboration process and product effectiveness, efficiency and satisfaction?"

is discussed in Bharadwaj's[3] collaboration patterns paper, wherein he explicitly discusses the impact changing collaboration processes have on the selection of groupware, and the evaluation that must be performed to determine suitable patterns for collaboration and thus dictate support software choice.

Of course, there are several other outstanding issues from the previous workshop that sadly were not addressed by contributors to this year's workshop. In particular the role of metrics and the selection of appropriate ones for evaluation, though mentioned by several of the papers this year were not discussed in depth.

1.1 Participation

The workshop this year had a smaller number of participants than usual; a complete list is available on the ECE Workshop Archive Page[6]. Despite the reduced attendance, delegates came from a wide range of disciplines including sociology, general computer science/software engineering and pure evaluation backgrounds.

Due to the reduced level of participation and a desire to increase interaction between the remaining participants, the organisers experimented with a new format for the ECE workshop this year. Instead of confining most of the discussion to a session held the day after the workshop as last year, a new format to increase the frequency and duration of discussion was tried. Papers studying similar areas were presented in pairs and immediately followed up with a discussion session of at least half an hour. These discussions took the form of panel sessions or free-form discussion, with a debate on all the issues at the end of the day. This aimed to promote two things, firstly comparative discussion of issues raised in the two papers while the topics were uppermost in delegates' minds and to avoid the monotony of an ordinary workshop, where due to time constraints a number of papers are consecutively presented to a largely passive audience. This is not engaging for the audience nor fair on the presenters of papers later in the day who may not have the audience's full attention.

2 Paper Summaries

Though each of these papers concentrated on different evaluation techniques and study subjects, there were several common factors. Firstly all the papers studied practical approaches to evaluation with a strong focus on results instead of the theoretical underpinnings of evaluation. Consequently, all the evaluations used common techniques though often in novel or modified ways. Secondly, many of the papers concentrated on embedding evaluation within the normal software lifecycle, to guide maintenance and provide an indication of project progress and success. Finally, all but one were concerned with practical evaluation of ongoing, or recently concluded projects.

The summary paragraphs below introduce the topic of each paper in the workshop. For a complete understanding of the research, the reader should refer to the original sources elsewhere in these proceedings.

An Enhanced Approach to Support Collaborative Systems Evaluation by Josie Huang[4]

Glasgow Caledonian University and the author of this paper were selected as the evaluation partner for the DIECOM project; a consortium of manufacturers and academic partners concerned with distributed configuration management with a focus on the automotive industry. This paper summarises the findings of the the final DIECOM evaluation and discusses the evaluation model and its impact on the developers. The model mirrored the software development lifecycle, with extra phases (follow-on) added in the enhanced model to allow improvement of the evaluation system. Users reported some of the same problems with the original evaluation framework (difficulty in applying the evaluation instruments) as they did with DIECOM itself (too many options in the toolset), showing that this framework is thoroughly integrated with both the software lifecycle and the software itself.

The workshop organisers also gave the Best Paper award to this author, as her work succintly addressed many of the explicit and tacit themes of the workshop. Additionally, the work included credible results from a detailed evaluation framework that will now be applied elsewhere.

Work Centred Evaluation Of Collaborative Systems - The COLLATE Experience by Hanne Albrechtsen et al[1]

The COLLATE environment applied collaboratory techniques to the domain of film research, specifically in the study of censorship decisions in the Fascist states of 20th Century Europe. Researchers from the Czech, German and Austrian national film archives collaborated using the environment to study two films: “Die Drei von der Tankstelle” a German satirical film censored by the Reich authorities and Battleship Potemkin, which needs no introduction. The users were then invited to an evaluation workshop and invited to provide feedback to the developers on the COLLATE system capabilities by means of a structured discussion. This paper discussed the evaluation workshop approach and gave samples of its findings with recommendations to the developers of COLLATE.

Modelling traceablity systems in Food Manufacturing Chains by Lucia Lo Bello et al[2]

EU regulation has been written to require traceability information to be inserted into the food chain by food processing companies. The traceability information will be used to track down shipments of contaminated or otherwise poor quality food before they end up on supermarket shelves. Given the rapid nature of the food production process today — in the test application grain was turned into pasta, packaged and shipped to retailers in less than a day — any traceability system must be essentially real-time with very stringent performance requirements.

Consequently this evaluation focussed mainly on the performance of the system and was constrained somewhat by the test bed. While the evaluators would have preferred to study other more interesting aspects of the system than response time, this was practically the only evaluation criteria the food consortium was interested in. Therefore future work to study the system in greater depth is planned.

An Evaluation Framework to Drive Future Evolution of a Research Prototype by David Nutter et al[7]

While standard evaluation involving user studies and complex manual instrumentation is very useful, when resources are limited performing ongoing evaluation in this way is difficult. One such example of resource-limited projects are research projects, where evaluation is usually restricted to the end of the funded research. This paper outlined a framework for performing a small-scale evaluation without a requirement for user involvement. Thus, developers can experience some of the benefits of proper evaluation with much less overhead. Consequently this evaluation may be used to guide future development of the project whereas evaluations isolated at the end of the project may not, though they will be more accurate.

However, at some point user-focussed rather than developer-focussed evaluation will become necessary. At this point, the knowledge gained from designing the developer-focussed evaluation will be useful in preparing the more comprehensive user-focussed evaluation.

Evaluating Adaptability in Frameworks that Support Morphing Collaboration Patterns by Vijayanand Bharadwaj et al[3]

Participants in collaborative projects adopt particular patterns of collaboration across the project lifecycle. This paper describes evaluation techniques for studying these patterns, providing a taxonomy of patterns and examining their adaptability. The effect that these patterns have on collaboration within projects is also discussed, for if the collaboration patterns are evolving the support software must also evolve for best results.

Therefore project planners would be well advised to conduct an evaluation of this type at various stages in their project so the most appropriate collaboration support software can be selected. At first, the intial heirarchy may be used as a guideline, however during the course of the project natural leaders in certain areas might emerge with a consequent shift in collaboration patterns to take account of the new user's expertise and power. If the collaboration software used does not support this leader in their assumed role (by excessively restricted permissions etc) the project will not perform so well.

The remainder of the paper discussed the evaluation of EkSarva, a system that aimed to support this adaptable behaviour by modifying defined workflows.

3 Summary of Discussion

3.1 Initial Discussion

Instead of the planned keynote, the workshop was opened by a group discussion. To start the discussion a short presentation was given and two questions posed to the delegates:

1. Information Systems design must consider strategies for adoption, realisation and evaluation from both technical and social viewpoints.

2. Evaluation like the system itself must be considered as continous and evolving activity as the system evolves and adapts over time.

Revisiting some old evaluation issues in the presentation was thought helpful by one delegate, in particular the issue of taxonomies. When officially mandated communication breaks down, participants in collaborations create their own taxonomies and collaboration processes by default: there is sometimes no benefit to imposing processes. There are two methods to study this; ethnographically or by studying the artefacts of the process. Taxonomies are however fluid, and must be kept up to date, invaluable aids to evaluation though they are.

Priorities differ between industry and academia, and this has implications for evaluation as it is often the first thing to be skipped if time is short. Consequently, tool support and simple evaluation are vital and care must be taken to select an appropriate evaluation method- concentrating on technical measures such as quality is not always appropriate as collaborative systems have social and economic aspects as well. The separation of development and evaluation teams often causes problems here; though an external perspective is useful, without cooperation between the two teams evaluators will have limited knowledge of their study subject and developers unsure how to implement evaluation recommendations. Additionally, research projects generally deal with unknown issues, whilst known issues are the preserve of business projects. Consequently, researchers feel the need to perform exploratory evaluation more than their business counterparts.

3.2 1st Paired Session

Immediately a comparison was made between the COLLATE paper[1] and the food traceability paper[2]. The former was an unregulated, almost unknown domain of work whereas the latter was a highly regulated, well-known domain. Though legal regulation is a strong driver of evaluation, specifics are rarely embedded in the law so evaluation practitioners must still select appropriate proxy measures and targets to determine compliance with the law. In the legally regulated domain, unambiguous quantitative data is therefore very important. However, extracting this sort of data from legacy systems such as food processing machinery is difficult, leading to concerns about adaptability of the target system impacting the success of evaluation.

Standards can assist here, but often standards (such as OPC for food processing) do not contain all the data necessary for successful evaluation. Moreover, agreeing a standard, or even sharing development of a common technology is extremely difficult.

The notion of data provenance is also important, even in the archival domain where information may have been tampered with. The problem for collaborative systems is that work done to data "offline" is untraceable, so COLLATE sticks to annotation only. However, users do like provenance. The highly regulated food traceability system relies on cryptographic provenance techniques from the ground-up to ensure records have not been tampered with and promote trust.

At this point the issue of scalability appeared; in systems where the collaborative object (foodchain) is frequently dynamic compared to a static collaborative object (film archive) the evaluation may have to change to encompass changes in the underlying collaboration, for example a food processor changing its grain supplier. Though change is possible in this domain, the relationships between participants are usually quite similar. Similarly, though film criticism is freeform within each of the archives involved in COLLATE, there are numerous rules governing collabo-

rations between them, thus changing the collaborative behaviour of the archivists and affecting the evaluation. The concern was expressed that small working studies may not necessarily scale to a large sample size.

Constraints imposed on the projects such as a limited test application involving only three food mills may have an impact on the success of the evaluation. Consequently, authors of both studies expressed a desire to do future work to address some of the omissions they made.

Finally, the contrast between the studies choice of evaluation methods (exclusively quantitative vs. exclusively qualitative) was highlighted. Both authors justified why they'd chosen the techniques by referring to the requirements (the law and existing archivist practices respectively) Moreover, emphasis was placed on the fact that requirements issues dictate the method of study design (top down or bottom up).

3.3 2nd Paired Session

The study of collaborative patterns[3] allows evaluators and managers to to discuss the system requirements and eventual evaluation of them in common business terms rather than technical ones. Identifying the real users of the system and getting them talking with managers about their collaborative needs is therefore a good way of improving the collaborative system's applicability to their needs and provides a neat business case for supporting evaluation activities. Extracting processes and collaboration patterns from business planning tools such as Microsoft Project was examined and the OPHELIA projectcited as a system which did this (albeit as an example).

Systems architecture was identified as having consequences for evaluation design, in particular selecting the features of a big system to evaluate and the adoption level of user tools were key issues here. Managing complexity by splitting big systems such as collaborative teaching environments into two or more pieces and devising appropriate evaluations for each can help here. It is interesting to note that complexity in evaluation can be technical (i.e. a complex piece of support software) or collaboration-related (i.e. a very complex, perhaps tacit process for collaboration). An example of the latter complexity would be the complex rules, sometimes unwritten, governing collaboration between film archives. With this kind of process, systems developers might well decide to leave coordination to the user (e.g. LUKSE)

This complexity management also prompted the intriguing idea of an evaluation component for each software component later in the discussions.

As a final remark, the problem of massively inflexible, top heavy systems such as Blackboard and MiX email systems were discussed and the problems they caused for evaluators. The consensus was that it was sometimes good to step back and revisit old controversies; the current trend is for customisable systems but many deployed collaborative environments cannot be modified to support evaluation (or anything else) without a major development effort.

3.4 Closing Remarks

Though a systematic approach to evaluation is considered to be the ideal solution, it is sometimes not feasible or desirable and certainly not a necessity for success.

The idea of "evaluation components" was raised here, with a question to Huang asking her to explain whether her integrated-lifecycle model of evaluation[4] could be used to assist in building an evaluation system for the world of toolbox systems. With slight modifications to the framework to take account of bits of code rather than people as actors in the evaluation process, she agreed that this would be possible. This lead onto an interesting discussion of how evaluation frameworks deal with change; in the case of the COLLATE evaluation, if the change affected user practices some aspects of the evaluation would need a rerun but if the software was modified this would not require a re-run as the goal of COLLATE was support of existing user practice rather than defining user practices from scratch to solve a particular problem.

Finally, the issue of user motivation for successful evaluation was raised, especially when unmotivated surrogate users such as students are used, potentially leading to biased results. The examination of this issue and how surrogates can be employed in evaluation (if at all) is a key issue for any future ECE workshops.

4 Conclusion

The outstanding issues from the workshop fell into three areas: social, technical and methodogical. Social issues are those ocurring because of the impact evaluation has on everyday business and academic activities; technical issues are those arising from the technical capabilities of the system under evaluation and methodological issues arise from the limitations of evaluation processes and toolsets.

The first social issue was that persuading managers and other decision makers that evaluation is important is difficult, and consequently evaluation efforts may not recieve the resources they require. This relates to another social issue: dialogue (or lack of it) between evaluation teams and developers where these teams are separate. These two issues have a common problem; lack of management buy-in to evaluation as a successful and effective support for software development. To address this issue, practitioners should examine ways of making the business case

for evaluation (e.g. satisfied users, higher quality software etc etc). The difference in world-view between researchers/evaluators and business people is also relevant: the former deals with unknowns and the latter with knowns.

Finally, the lack of real-world study subjects is a key issue for evaluation researchers; while "making do" with fellow researchers and students is sometimes satisfactory, efforts to enroll real users are vitally important. This year's workshop was interesting as three out of five paper presenters brought results from real-world industrial evaluation, compared to the previous ECE workshop where only three out of seven papers had real results.

A key technical issue is system scalability. From an evaluation perspective, it is desirable to involve as many real users as possible and get them all using the system at the same time in order to study their interaction. However, many pure-research systems are insufficiently mature and cannot support large numbers of concurrent users. Therefore, system scalability has an effect on study scalability. Different system designs also have an impact; for example highly-integrated monolithic systems (e.g. Microsoft Exchange) have different evaluation requirements from "toolbox" systems (e.g. standard e-mail) where users may select from a range of collaborative tools to meet their needs. While tightly integrated systems may lead to complex, unwieldy evaluation methods which attempt to study the effect of each feature in the system, toolbox-type systems require at least some users to adopt each tool in order to evaluate their effect; difficult with a limited userbase. Adaptability is also critical, as systems may need to be modified to support evaluation (e.g. automated data collection) or as a result of evaluation (e.g. the users studied find the system ineffective). All these technical issues pose an intersting question: can we develop evaluation "components" to match our software components? Thus, constructed an evaluation framework for a collaborative application will mirror constructing the application itself; each evaluation "component" will be added to the framework as its corresponding software component is integrated into the system under development.

Methodogical issues identified by the workshop include the requirement for a positive evaluation result to indicate project success. Alongside checking that a project meets its goals, such evaluation must provide pointers for future work. Therefore, there is a need for meta-evaluation to study the effectiveness of various evaluation methodologies in this area. The papers in the workshop generally concentrated exclusively on either empirical measures or qualitative assessment techniques such as case studies and interviews. The overall aims of the evaluation caused this strong contrast: evaluations concerned with requirements capture etc favoured qualitative techniques while those concerned with showing steady project improvement or overall success preferred quantitative measures. However, this issue of technique applicability deserves further study in light of the need for meta-evaluation identified earlier and the issue of blending qualitative and quantitative techniques deserves examination.

Finally, the constraints of the test application imposed on certain types of project may impact the scope and success of the evaluation. Even if the evaluation design is good and the collaborative system sufficiently scalable, if the test application is limited to trivial matters the evaluation result will be of limited value. Therefore, the requirements of successful evaluation should be taken into account when drafting research project proposals to ensure that the evaluation cran produce meaningful results.

4.1 Recommendations

Both bottom-up and top-down study designs have their uses; the former when a process or system to evaluate already exists and quantitative measures can be used to study it directly whereas the top-down approach is necessary when user studies to elicit system requirements are useful. In an ideal world of course, all systems would be subject to both these types of evaluation at various stages in their lifespan. However, a systematic approach to evaluation is not always feasible, often due to lack of resources, or even desirable if the system is in flux. In the latter case, exploratory evaluation to discover things about the system is useful, whereas for stable systems systematic evaluation if possible is best.

Integrated evaluation has a role in each phase of the software lifecycle: how else does one decide whether a particular phase in development was a success or failure? Furthermore, if following an iterative model of software development, output from evaluation efforts may be used to improve the next iteration and thus improve project performance.

Finally, the key recommendation endorsed by all participants was to keep evaluation instruments, tools and process simple, so that users can participate without expending large amounts of time on understanding and enacting the evaluation. An evaluation should be as lightweight and focussed as possible to avoid user and managerial resentment.

4.2 Looking forward

In the closing session of the main conference, a proposal was made to return WETICE to its interactive, cross disciplinary roots rather than allowing it to assume a more standard conference format. In particular, the Enterprise Security workshop will be holding Joint Sessions with others next year. However, the recurring theme in ECE this year was the need for integrated, ongoing evaluation during the software lifecycle with dialogue between development

and evaluation teams. Evaluation is not a field distinct from others; every researcher and practitioner must evaluate their output in some way though they may not apply formal techniques to do so. Indeed, when the author attended other workshops in the main conference it was noticeable that many of the paper presentations had a slide or two on evaluation issues tucked away at the end. Consequently, the ECE team would like to encourage authors in other workshops to submit full or short papers discussing any evaluation component of their ongoing work to next year's ECE workshop and not to leave evaluation as a footnote to other research!

References

[1] H. Albrechtsen, H. H. Andersen, and B. Cleal. Work-Centred Evaluation of Collaborative Systems - the COLLATE Experience. In WETICE '04 [8]. http://hemswell.lincoln.ac.uk/wetice04/.

[2] L. L. Bello, O. Mirabella, and N. Torrisi. Modelling traceablity systems in food manufacturing chains. In WETICE '04 [8]. http://hemswell.lincoln.ac.uk/wetice04/.

[3] V. Bharadwaj, Y. R. Reddy, S. Kankanahalli, S. Reddy, S. Selliah, and J. Yu. Evaluating adaptability in frameworks that support morphing collaboration patterns. In WETICE '04 [8]. http://hemswell.lincoln.ac.uk/wetice04/.

[4] J. P. Huang. An enhanced approach to support collaborative systems evaluation. In WETICE '04 [8]. http://hemswell.lincoln.ac.uk/wetice04/.

[5] J. Newman, E. M. Raybourn, and J. P. Huang. Wetice2003 evaluating collaborative enterprises workshop report. pages 131–136, Linz, Austria, June 2003. IEEE.

[6] D. Nutter. ECE workshop archive page. Web, June 2004. Link valid 28/6/2004.

[7] D. Nutter, C. Boldyreff, and S. Rank. An evaluation framework to drive future evolution of a research prototype. In WETICE '04 [8]. http://hemswell.lincoln.ac.uk/wetice04/.

[8] *13th IEEE international Workshops on Enabling Technologies For Collaborative Enterprises (WETICE)*, Modena, Italy, June 2004. IEEE Computer Society. http://hemswell.lincoln.ac.uk/wetice04/.