AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

Prototyping for Requirements Elicitation and Validation: A Participative Prototype Evaluation Methodology

Ann Hickey
University of Arizona

Douglas Dean
University of Arizona

Prototyping for Requirements Elicitation and Validation: A Participative Prototype Evaluation Methodology

Ann M. Hickey
Douglas L. Dean
Center for the Management of Information
University of Arizona

Abstract

Prototyping is widely recommended as an excellent mechanism for requirements elicitation and validation. However, few details are available on how prototypes should be used for this purpose, especially in a group environment. The goal of this research is to develop and evaluate a methodology for using prototyping to elicit and validate requirements from large user groups. An initial prototype evaluation methodology was developed, assessed during a pilot case study, and revised to support different evaluation phases and types. The revised participative prototype evaluation methodology provides a specific structure for each prototype evaluation phase with detailed methods and GroupSystems tools defined for each evaluation type. An overview of the face-to-face procedures by evaluation type is included in this paper.

Introduction

Over the last 20 years, many systems development methodologies have been proposed to address the problem of identifying user requirements. However, these methodologies generally focus on the analysis of user requirements rather than the elicitation of those requirements from the users. They also make an implicit assumption that users know and can articulate their requirements – possibly with the help of an analyst. The seemingly endless reports of spectacular system failures and systems delivered to users only to be rejected as “not what they need” indicate that this assumption is not valid. Research has shown that many users have difficulty articulating their requirements until they see them. As Frederick Brooks states, “… it is really impossible for a client, even working with a software engineer, to specify completely, precisely, and correctly the exact requirements of a modern software product before trying some version of the product” (Brooks, 1986; Cardenas, Jianhui, & Zelkowitz, 1992, p. 29). Prototypes have been proposed as an excellent mechanism for helping users get past this barrier by providing a product to evaluate (Conger, 1994; Davis, 1993; Davis, 1995; McConnell, 1996; Schach, 1996). However, very little has been written about how this evaluation should be performed, especially by groups of users simultaneously.

The purpose of this research is to develop and evaluate a methodology for the use of prototyping for eliciting and validating requirements from large groups of heterogeneous users. Since the purpose of requirements prototyping is to get feedback from multiple users, a participative prototype evaluation methodology is key to a successful prototyping process. However, most prototype evaluations simply provide users access to the prototype and ask for their feedback. Little or no structure is provided. Thus, feedback is often hit-or-miss. Developers may also have difficulty reconciling the often-conflicting feedback from multiple users. Electronic Meeting Systems, such as GroupSystems, have been very effective in collecting information from user groups and therefore should be extremely useful in supporting participative prototype evaluations. Extensive GroupSystems experience has shown that the more appropriate the meeting process, facilitation support and tools, the more the meeting. These observations lead to the primary research question: *What methodology best supports effective user evaluation of requirements prototypes in a group environment?*

The initial participative prototype evaluation approach is summarized next, followed by results and discussion of a pilot case study. The final evaluation methodology is then summarized. Finally, the conclusion highlights contributions to research and practice.

Participative Prototype Evaluations

The goal of participative prototype evaluations is to produce user requirements feedback using an efficient and effective group process that keeps users focused on the essential requirements embodied in the prototype versus the prototype’s implementation of those requirements. To achieve this goal, a multi-step process was used to develop the basic approach. First, the types of requirements that could be captured from users during evaluations were identified. These factors were then used to develop generic questionnaires that could be used to prompt users to evaluate specific requirements factors. Finally, the questionnaires were incorporated into a three-phase prototype evaluation strategy.

Because the evaluation factors and questionnaires must be tailored to the specifics of a particular prototype to efficiently gather the maximum amount of information the first evaluation phase is the *Preliminary Evaluation and Planning Phase*. In this phase, the development team and key functional representatives conduct a generic evaluation using the standard questionnaires.

This evaluation provides some feedback to the developers, but, more importantly, helps the developers understand the evaluation process so that they can adequately tailor the process and the questionnaires for the second phase of evaluation — the *Face-to-Face User Evaluation Phase*. A face-to-face evaluation addresses the following items:

1. Meeting introduction to explain the purpose and focus of the evaluation to the users.
2. Prototype overview and user training.
3. Scenario evaluation where users execute specific scenarios and evaluate the prototype using the tailored questionnaire.
4. Summary prototype evaluation where users explore the prototype in an unstructured manner, recording their evaluations using the comprehensive system questionnaire. The goal of these two steps is to elicit new/missing requirements using the questionnaires as a checklist to ensure users consider all requirements.
5. Joint user review, analysis, integration, and prioritization of user assessments and requirements identified during the prototype evaluation.

The third phase is a *Distributed User Evaluation Phase*. This type of evaluation is limited to scenario evaluations, questionnaire completion, and unstructured user comments. Although a distributed evaluation lacks the richness of the face-to-face user evaluation and provides no mechanism for integrating the evaluation results, the distributed evaluation can be a cost-effective approach as long as some mechanism is provided for resolving conflicting evaluations.

Results

The proposed participative prototype evaluation approach was assessed during a pilot case study of two Department of Defense prototype evaluations. The evaluations were conducted with a small group of approximately eight decision-makers, developers, and users. As anticipated, there were some problems with the prototype evaluation caused by the small group, its mix of decision-makers and users, and the limited time available because of scheduling constraints. However, even with these problems, the prototype evaluations clearly showed the power of prototyping for requirements elicitation and provided a valuable learning experience for the evaluation process itself. The results of these evaluations are summarized next.

The average rating on all evaluation questions was 3.30 (of 5) for the first prototype and 3.48 for the second with 46 requirements identified for the first prototype and 41 requirements for the second. A detailed review of the ratings and requirements showed consistent evaluations by users, i.e., low ratings were usually accompanied by several important requirements problems. High ratings had fewer or less important requirements identified.

The major problem experienced during the evaluation was that the cognitive load on the users seemed to be excessive. Users were expected to learn the prototype system, execute a scenario using that system, learn the GroupSystems tools for recording their comments, and switch between the prototype and GroupSystems as they were evaluating the prototype. Users also had a difficult time identifying the appropriate question to record their comments. In fact, during some phases of the evaluation, users resorted to verbal discussions and did not record their comments in the questionnaire at all.

A second problem was that evaluation process switched between demonstrations and scenario walk-throughs by developers and the planned user evaluation of the scenarios and prototype. In many cases, it appeared that the users did not have the expertise to evaluate the prototype on their own or that the prototype was not complete enough for them to do so.

Discussion

The lessons learned during the pilot case study are briefly summarized below.

1. *Process Simplification.* The main lesson learned from the prototype evaluations was that the process needed to be greatly simplified. The overall cognitive load needed to be significantly reduced.
2. *Technographer Support.* The addition of a technographer to capture requirements during verbal discussions would also be extremely helpful.
3. *Reduced Evaluation Structure.* Most importantly, the evaluation methodology needed to be refined to support the three phases of group problem solving: divergence, convergence, and consensus building. The initial approach jumped directly to convergence by using a highly structured questionnaire to guide the evaluation. Users seemed very uncomfortable with this degree of structure during the early evaluation phases. They seemed to prefer a divergent, free-form system evaluation process before beginning structured scenario evaluations.
4. *Tailored Evaluations.* It also became apparent during the evaluations that to maximize process efficiency and effectiveness, the evaluation activities must be carefully tailored to the maturity of the prototype and level of user familiarity with that prototype.
5. *Required Evaluation Time.* Most importantly, the facilitator must allow sufficient time for the entire evaluation session, especially for the assessment and joint integration of any proposed requirements. Requirements integration and validation is the primary benefit of face-to-face evaluations.

Participative Evaluation Methodology

The final participative prototype evaluation methodology addresses the concerns identified during the pilot study. It recommends a specific structure for each prototype evaluation phase with detailed methods and supporting GroupSystems tools defined for each evaluation type. Only an overview of face-to-face procedures by evaluation type is included next due to the space limitations of this paper.

Demonstration-Based Prototype Evaluations. When developers are first introducing a prototype to a new user group or if the prototype is not yet fully functional, a demonstration-based prototype evaluation may be preferred. In this type of evaluation, the developer demonstrates the prototype with the results displayed on the public screen. During the demonstration, individual users discuss the prototype, ask questions, and record their comments using GroupSystems. A technographer records any requirements-related verbal comments that are not recorded directly by the participants.

Chauffeured Scenario Prototype Evaluations. After users have seen a demonstration of the prototype and had some hands-on practice, they are ready to begin scenario evaluations. However, because they are still fairly unfamiliar with the system, a chauffeured scenario evaluation where the developer walks the users through the scenario evaluation is recommended. Users execute the scenarios at their own computer. To reduce the cognitive load of running the scenario and then switching to GroupSystems to record comments, two-person teams are recommended. One user runs the prototype to try the scenarios as guided by the developer. The other user runs GroupSystems to record the team's comments. Again, the technographer ensures that verbally discussed requirements recorded.

Independent Scenario Prototype Evaluations. In this type of evaluation, the users are sufficiently knowledgeable and the prototype sufficiently mature so that user teams can independently evaluate specific scenarios. The developer briefly describes the scenarios and provides training if required. The two-user teams then independently execute and evaluate each scenario. The developer answers user questions one-on-one. Since there will be less verbal discussion during this evaluation type, a technographer is generally not required.

Comprehensive Prototype Evaluations. Following the scenario evaluations, the two-user teams independently evaluate the complete prototype. The developer should be available to answer questions one-on-one as they occur during these evaluations. Again, one user runs the prototype in unstructured mode, trying whatever features interest him or her. The other user records the team's comments in GroupSystems. Finally, the team completes the tailored prototype evaluation questionnaire.

Conclusion

The benefits of prototyping for requirements elicitation and validation are well known. This research has focused on developing and assessing a participative prototyping evaluation methodology to permit development teams to gain those benefits through efficient user participation in the evaluation process. For researchers and practitioners, the research highlighted that distinct phases and types of prototype evaluations require different meeting methods, structure, and GroupSystems tools. The proposed methodology provides a specific structure for each prototype evaluation phase and defines detailed methods and supporting GroupSystems tools for each evaluation type which practitioners can use now to conduct their own prototype evaluations.

Future research includes further refinement of the methodology based on additional case studies and integration of this participative approach to prototype evaluations into the Collaborative Software Engineering Methodology (Dean, Lee, Pendergast, Hickey, & Nunamaker, 1997-98) designed to maximize user involvement throughout the entire systems development process.

References

References available upon request from the first author (ahickey@cmi.arizona.edu).