

8-15-1997

# Group Decision Support for Software Requirements Analysis

Fatemeh Mariam Zahed

*University of Wisconsin Milwaukee, zahedi@csd.uwm.edu*

Padmal Vitharana

*University of Wisconsin Milwaukee*

Follow this and additional works at: <http://aisel.aisnet.org/amcis1997>

---

## Recommended Citation

Zahed, Fatemeh Mariam and Vitharana, Padmal, "Group Decision Support for Software Requirements Analysis" (1997). *AMCIS 1997 Proceedings*. 131.

<http://aisel.aisnet.org/amcis1997/131>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISEL). It has been accepted for inclusion in AMCIS 1997 Proceedings by an authorized administrator of AIS Electronic Library (AISEL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# Group Decision Support for Software Requirements Analysis

Padmal Vitharana and [Fatemeh "Mariam" Zahedi](#)

Names are in alphabetical order

University of Wisconsin Milwaukee

P.O. Box 742 Milwaukee WI 53201

contact: [zahedi@csd.uwm.edu](mailto:zahedi@csd.uwm.edu) or (414) 229-6454

## Abstract

The importance of Requirements Analysis (RA) in building quality software systems is well documented. However, the involvement of various individuals in RA including users and developers creates numerous communication difficulties in correctly identifying user requirements. The purpose of this research is to formalize the stages of RA and identify methodologies appropriate for each stage, incorporating components of group decision support systems (GDSS) that exploit the group dynamics in facilitating and improving the process. We develop a prototype of this GDSS and test its impact on the quality of RA.

Requirements analysis (RA) plays a critical role in building quality software. RA is the process of identifying a user's needs and determining what to build in a system [Holbrook 1990]. It has been shown that defects injected into software during RA are costlier to correct than those injected during subsequent phases of the development life cycle [Fairley 1985; Schneider et al. 1992]. Research has also shown that many system failures can be attributed to the lack of clear and specific RA [Cooper and Swanson 1979; Davis 1982]. The financial consequence of RA has been noted by Boehm [1981] and Mittermeir et al. [1982]. Taggart and Tharp [1977] have reported on the awareness of senior managers of the critical impact of RA in design.

Kan et al. [1994] emphasize the importance of clearly identifying customers' wants and needs, and systematically eliciting their requirements. There are a number of methodologies designed to facilitate RA, categorized in concrete and abstract methods [Gutierrez 1989; Zahedi 1995]. The concrete methods include object-oriented approach, games, simulation, prototypes, pilots, and operational experiments. The abstract methods include Delphi, surveys, and repertory grids. However, none of these methodologies fully considers the process involved in RA or its inherent group nature.

Zahedi [1995, p. 156] discusses the RA process and divides it into: elicitation, anticipation, verification, and validation of users' needs. Bostrom [1989] recognizes the group dynamics of RA and emphasizes the importance of communications among the diverse group of users. Valusek and Fryback [1987] classify communications barriers to a successful RA into three categories: within, between, and among. "Within" communication difficulties arise due to individual cognitive limitations. "Between" obstacles are in communications between users and analysts. "Among" communication barriers are a combination of both within and between, and entail the problems due to variety of users' needs and points of view. Bostrom [1989] suggests that a good RA approach should attempt to reduce "among" obstacles in the process.

In this research, we design a group decision support system (GDSS) for RA in order to facilitate the group process and reduce the obstacles. In doing so, we formalize the requirements gathering process by modeling the phases of the RA process, and identify the group decision support component needed to facilitate the accomplishment of objectives in that phase.

In developing the system, we divide the RA process into nine phases as shown in Figure 1. In each phase, we identify the objective of the group work, the participants, and the GDSS component that is most appropriate for accomplishing the objective of that phase.

**1. Requirements elicitation.** In this phase, the users' needs are to be identified. Objective of this phase is to create a raw list of what the users think are important requirements. The group involved in this phase consists of various types of systems users and customers [see Zahedi 1995 for a review of constituents of a software system]. The GDSS component for this phase is the brainstorming method.

**2. Requirements reduction.** This phase eliminates redundant requirements, corrects any misperception that may exist regarding the capabilities of the existing software systems (if the analysis is done for upgrading the system), and adds anticipated needs. The objective of this phase is to create a concise list of users requirements. The participants of this phase are the users team and technologists who serve as consultants. The GDSS components for this phase are reducing and ranking techniques. More elaborate GDSS could be the "elimination by aspects" method [Tversky 1972] and "conjunctive decision making," [Todd and Benbasat 1992]. These methods allow the users to reduce the list of requirements to those which are considered important.

**3. Requirements categorization.** In preparation to translate the elicited and reduced requirements to technical specification, one needs to group and categorize them into appropriate levels of generality and similarity. The objective of this phase is to produce a categorized list of users' requirements. The participants in this phase are users as well as developer-selected representatives. The knowledge of these representatives helps the user group to identify categories for grouping the requirements. The GDSS tools for this phase are categorization and creation of a requirements hierarchy---starting from the most general level and moving down the hierarchy to more detailed categories. For example, "user friendly" could be a requirement at the general level, below which one may have more specific requirements such as better help, friendlier screens, and fewer number of screens to travel.

**4. Requirements evaluation.** In this phase, the categorized requirements are evaluated. The objective of this phase is to come up with priorities or relative ratings for requirements. The participants in this phase are users and representatives of the developers. The GDSS component for this phase is group AHP [Aczel and Saaty 1983; Zahedi 1995]. In this method, the group produces relative ratings that prioritize the requirements.

**5. Requirements to technical translation.** In this phase, the users' requirements are to be translated to technical terms. The objective of this phase is to map the users' requirements to system terms for precise specifications. The participants in this phase are the developers team and representatives of the users. The GDSS component of this phase is the house of quality in quality function deployment (QFD) [Zultner 1990; Zahedi 1995].

**6. Requirements to technical translation evaluation.** In this phase, the mapping of requirements to technical terms is evaluated. The objective of this phase is to produce a metric to measure the strength of relationship between the translation from users' requirements to technical requirements. The participants in this phase are developers and users' representatives. The GDSS tool for this phase is the correlation matrix of QFD.

**7. Technical requirements prioritization.** In this phase, the technical requirements are prioritized. The objective is to produce a prioritized list of technical requirements. The GDSS component of this phase is the priority computation in QFD.

**8. Requirements specification documentation.** In this phase, more detailed technical requirements are generated. The objective is to create enough technical details to write the specification document. The participants in this phase are the developers team and technical writers whose job is to produce the specification document. The GDSS component of this phase consists of second to n<sup>th</sup> level QFD [Zahedi 1995].

**9. Requirements inspection.** In this phase, the requirements are verified and validated using a formal software inspection process. The objective of this phase is to create a finalized requirements-specification

document that is unambiguous, clear, concise, and ready to be used for design and contracts. The participants in this phase are developers' and users' representatives, who actively participated in stages 1 through 8. Software inspection includes three main steps: individual preparation, group meeting, and rework [Fagan 1976]. The GDSS component provides the facility to capture comments in the individual preparation step, help identify defects from consolidated comments in the group meeting step, and facilitate the correction of defects in the rework step. An optional tool is the creation of a prototype from specification documents (phase 8) to aid the process.

The contribution our study is in formalizing the stages of RA and defining GDSS-based methodologies appropriate for each stage, in order to facilitate and improve the quality of RA. In this research, we plan to create a prototype for the design of GDSS for RA based on the above components, and test its performance in reducing process time, improving quality, and increasing the satisfaction of participants with the process of requirements analysis.

### References

- Aczel, J. and Saaty, T. 1983. "Procedures for synthesizing ratio judgments," *Journal of Mathematical Psychology*, Vol. 27, pp. 93-102.
- Boehm, B. 1981. *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ.
- Bostrom, R. 1989. "Successful application of communication techniques to improve the systems development process," *Information and Management*, Vol. 16, No. 5, pp. 279-295.
- Cooper, R. and Swanson, E. 1979. "Management information requirements assessment: The state of the art," *Data Base*, Vol. 10, No. 2, pp. 5-16.
- Davis, G. 1982. "Strategies for information requirements determination," *IBM System Journal*, Vol. 21, No. 1, pp. 4-30.
- Fagan, M. 1976. "Design and code inspections to reduce errors in program development," *IBM Systems Journal*, vol. 15, No. 3, pp. 182-211.
- Fairley, R. 1985. *Software Engineering Concepts*, McGraw-Hill, New York, NY.
- Gutierrez, O. 1989. "Experimental techniques for information requirements analysis," *Information and Management*, Vol. 16, pp. 31-43.
- Holbrook, H. 1990. "A Scenario-based methodology for constructing requirement elicitation," *ACM SIGSOFT Software Engineering Notes*, Vol. 15, No. 1, pp. 95-104.
- Kan, S., Basili, V., and Shapiro, L. 1994. "Software quality: An overview from the perspective of total quality management," *IBM Systems Journal*, Vol. 33, No. 1, pp. 4-19.
- Mittermeir, R., Hsia, P., and Yeh, R. 1982. "Alternatives to overcome the communication problem of formal requirements analysis," in *Requirements Engineering Environments*, M. Ohno (ed.), North-Holland, Amsterdam.
- Schneider, G., Martin, J., and Tsai, W. 1992. "An experimental study of fault detection in user requirements documents," *ACM Transactions on Software Engineering and Methodology*, Vol. 1, No. 2, pp. 188-204.

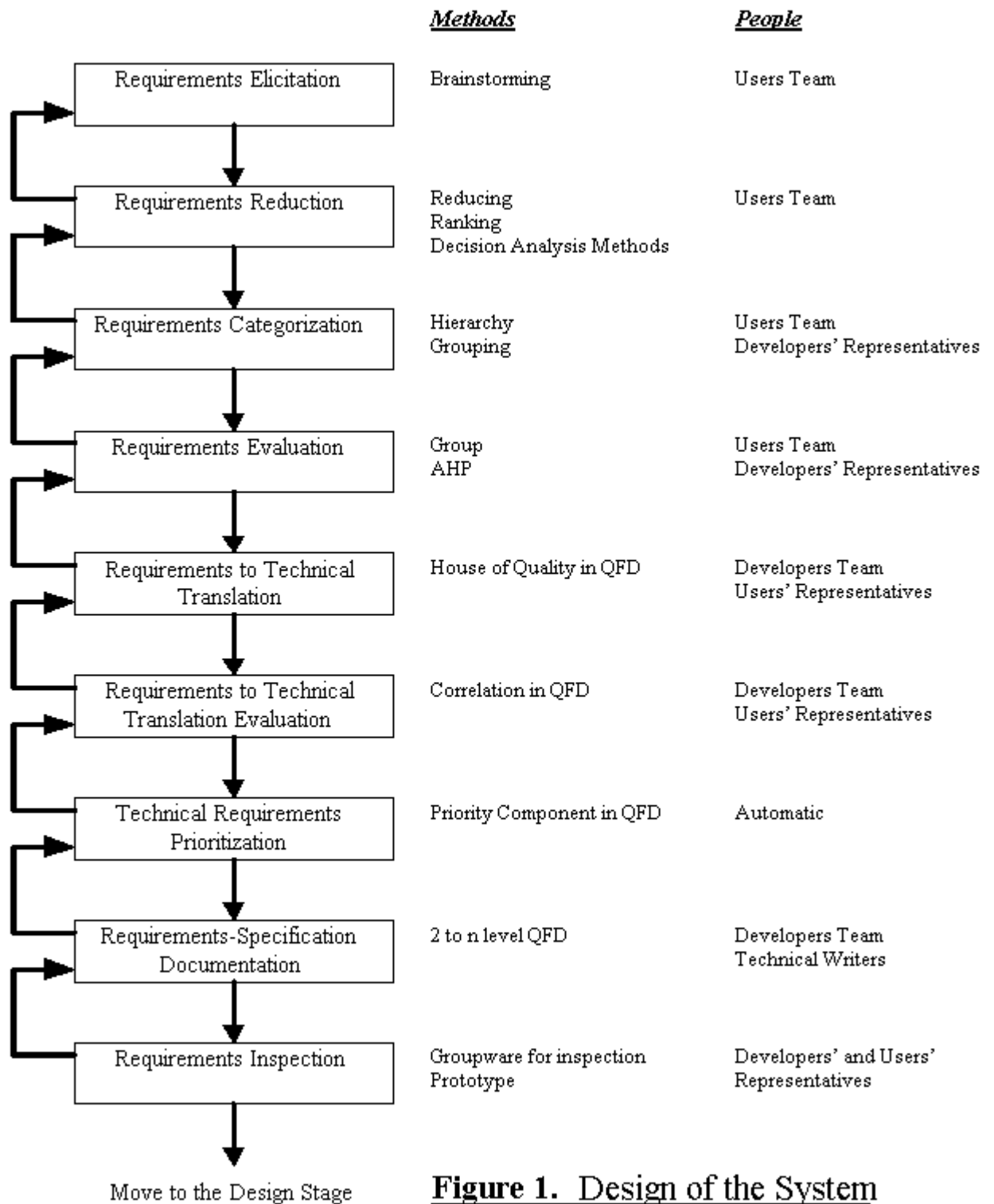
Taggart, W. and Tharp, M. 1977. "A Survey of information requirements analysis techniques," *Computing Surveys*, Vol. 9, No. 4, pp. 273-290.

Todd, P. and Benbasat, I. 1992. "The use of information in decision making: An experimental investigation of the impact of computer-based decision analysis," *MIS Quarterly*, Vol. 16, No. 3, pp. 373-393.

Tversky, A. 1972. "Elimination by aspects: A theory of choice," *Psychological Review*, Vol. 79, No. 4, pp. 281-299.

Valusek, J. and Fryback, D. 1987. "Information requirements determination: Obstacles within, among, and between participants," in *Information Analysis: Selected Readings*, R. Galliers (ed.), Addison-Wesley, Reading, MA, pp. 139-151.

Zahedi, F. 1995. *Quality Information Systems*, Boyd and Fraser Publishing Co. Danvers, MA.



**Figure 1. Design of the System**

