Information Requirements Analysis and Multiple Knowledge Elicitation Techniques: Experience with the Pricing Scenario System

Dong-Gil Ko
donkgo@pitt.edu
Management Information Systems, Katz Graduate School of Business
University of Pittsburgh, Pittsburgh, Pennsylvania 15260

Abstract

It is well understood that large percentage of software development costs are incurred during the earlier phases of the software development process, namely the information requirements analysis. Given the importance of information requirements analysis during software development process, it is surprising that there is limited research in this area to advance the knowledge for better equipping information systems project managers and analysts.

This paper examines a software prototyping project held by one organization through use of structured interviews, group sessions, and scenario analysis techniques. As suggested by Davis and Byrd et al., combining multiple elicitation techniques at more than one level in the organization allowed explicit and tacit knowledge to be surfaced. A discussion of gathering information requirements by tapping into upper management, middle managers, and end-users’ knowledge is provided leading to a development of an effective information system.

1. Introduction

It is well understood that large percentage of software development costs are incurred during the earlier phases of the software development process, namely the information requirements analysis [1, 2, 12, 13]. It is also possible, however, that much of the costs can be incurred during the later stages of the software development process (e.g., testing, implementation, etc.). This is attributable primarily to incomplete or poor analyses of user information needs conducted during the information requirements analysis. The importance to the development of effective information systems begins with information requirements that are complete and thorough, as evidenced by studies conducted by researchers in the field [4, 17]. Otherwise, the software development can experience a “domino” effect, whereby an error, or undetected or misdirected information requirements, can carry over throughout the development process. Implications include (1) increase in costs, (2) delay in project completion, (3) frustration among the developers and the end-user community, (4) potential changes in project scope, and (5) incomplete and less effective information systems.

Given the importance of information requirements analysis phase, it is surprising that there is limited research in this area to advance the knowledge for better equipping information systems (IS) project managers and analysts [8]. Byrd and his colleagues [3] provide a nice review of various elicitation techniques used in information systems development and call for a need to examine these techniques more rigorously. Specifically, they suggest research focused on combining two or more elicitation types to determine whether or not it is beneficial to use two or more techniques for gathering information requirements. Furthermore, Davis [6] suggests research examining information requirements gathered at two levels, organizational and application/system levels, in order to design and implement information systems. It is to this end that this paper is addressed.

Given the various knowledge acquisition techniques offered by Byrd et al., this paper focuses on “scenario” technique (scenario technique, scenario-based, and scenario analysis are used interchangeably) with some emphasis on structured interviews to achieve the following: 1) determine whether or not it is beneficial to use more than one information requirements gathering technique and 2) propose a general model for eliciting information requirements through scenario technique. Definition of “scenario” in this paper extends the
definition offered by Byrd et al. and is defined here as (1) some future desirable state(s) for identifying and understanding the events required to get there and (2) some current state(s) in which the firm conducts their business.

Scenario analysis has been gaining much attention in other disciplines such as computer science and strategy. Furthermore, some companies are exercising scenario techniques today, which lends itself to examination at the early stages of its development. The findings reported here challenges some of the common practices of conducting information requirements analysis. A company exercising scenario analysis in their prototyping effort is illustrated, and the value of the technique in developing an effective information systems is discussed. An initial finding is that the use of scenario analysis as a technique for gathering information requirements surfaced most of the tacit knowledge held by individuals (not necessarily all experts). This paper provides a brief overview of knowledge acquisition/creation and scenario analysis technique. This will be followed by an examination of a company exercising scenario analysis in their prototyping effort to illustrate the value of the technique in developing an effective information systems.

2. Knowledge Acquisition/Creation

Knowledge is a multifaceted concept with multiple meanings. Many researchers define the term knowledge in their own rights. Knowledge has been defined as information combined with experience, context, interpretation, and reflection [5], more costly to transfer than information [10], and most strategically-important resource which firms possess [16]. In Dretske’s [7] words:

"Information is that commodity capable of yielding knowledge, and what information a signal carries is what we can learn from it" (p. 44). “Knowledge is identified with information-produced (or sustained) belief, but the information a person receives is relative to what he or she already knows about the possibilities at the source” (p. 86).

Similarly, Nonaka [14] states that “information is a flow of messages, while knowledge is created and organized by the very flow of information, anchored on the commitment and beliefs of its holder” (p. 15).

Drawn from Michael Polanyi’s [15] assertion that individuals know more than they can explain, there are two distinct types of knowledge - explicit knowledge and tacit knowledge. Explicit knowledge (codified knowledge or declarative knowledge) represents knowledge that can be transmitted through formal, systematic language at low cost [10, 15]. Tacit knowledge (uncodified knowledge or procedural knowledge) represents knowledge that is “deeply rooted in action, commitment, and involvement in a specific context” (p. 16) [14] and transmitted at high cost [10].

<table>
<thead>
<tr>
<th>Tacit Knowledge</th>
<th>Explicit Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>Socialization</td>
</tr>
<tr>
<td>To</td>
<td>Internalization</td>
</tr>
</tbody>
</table>

Figure 1. Modes of Knowledge Creation [14]

Fundamentally, knowledge is created and held by individuals, and Nonaka [14] suggests that there are four different patterns of interaction between tacit and explicit knowledge (Figure 1). The assertion is that these patterns represent ways in which existing knowledge can be converted into new knowledge. Socialization represents a mode in which tacit knowledge can be converted into tacit knowledge through interaction between individuals, whether it is through language, observation, imitation, or practice. Externalization represents a mode in which tacit knowledge can be converted into explicit knowledge through interaction between regular individuals and expert individuals (based on contexts) whereby the regular individuals are the recipients of the knowledge. Internalization represents a mode in which explicit knowledge can be converted into tacit knowledge through individual learning, experimenting, and acting. Lastly, combination represents a mode in which explicit knowledge held by individuals can be converted into explicit knowledge through exchange mechanisms or reconfiguring of existing information (e.g., sorting, recategorizing, etc.). (For fuller and complete discussion on these four patterns of interaction between tacit and explicit knowledge, see [14]).

Nonaka’s four modes of the knowledge creation is very appropriate for this paper. From the perspective of those gathering the requirements, knowledge is created by converting and transferring explicit and tacit knowledge. Nonaka also suggests that the most effective mechanism for gathering information requirements is if the firm experiences all four modes of the knowledge creation - from tacit knowledge to both tacit and explicit knowledge, as well as from explicit knowledge to both tacit and explicit knowledge. Consistent with Byrd et al.’s study, the firm under study used scenario analysis, structured interviews, prototyping, and group observations to accomplish the goal of gathering
Table 1. Empirical Studies on Scenarios

<table>
<thead>
<tr>
<th>Authors</th>
<th>Year</th>
<th>Research</th>
<th>Unit of Analysis</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walz et al.</td>
<td>1993</td>
<td>observation software</td>
<td>software design team</td>
<td>• much information was presented to the team and never captured</td>
</tr>
<tr>
<td></td>
<td></td>
<td>design</td>
<td></td>
<td>• requirements determination seemed to “shut down” based more on project timing rather than on achieving a full understanding of the requirements</td>
</tr>
<tr>
<td>Zmud et al.</td>
<td>1993</td>
<td>laboratory experiment</td>
<td>requirements analysis protocol</td>
<td>• moderate support favoring the scenario-based techniques in eliciting information items from users facing ill-structured task contexts</td>
</tr>
<tr>
<td>Hsia et al.</td>
<td>1994</td>
<td>sample application</td>
<td>PBX system</td>
<td>• formal method for scenario analysis includes scenario elicitation, formalization, verification, generation, prototype generation, and scenario validation</td>
</tr>
<tr>
<td>Kazman et al.</td>
<td>1996</td>
<td>case study</td>
<td>software architecture analysis method</td>
<td>• use of scenarios enhanced both team communication and communication between the team and upper-level management</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• the level of structural description is dictated by the scenarios</td>
</tr>
</tbody>
</table>

information requirements through the four modes of the knowledge creation. What, then, is scenario analysis? This question is addressed in the following section.

2. Scenario Analysis

Scenario analysis is a technique and a process of gathering input from individuals, understanding that input, and analyzing that input to describe expected or anticipated system behavior [9, 11]. Scenario analysis differs from other techniques and processes because information requirements stem from current states in which the firm conducts their business as well as some future desirable states for identifying and understanding the events required to get there. Each scenario developed identifies a particular business, outlines step-by-step procedure for conducting a business, makes explicit a particular business allowing all parties involved to confirm or modify with ease, ensures that multiple ways of conducting a particular business are captured, and simplifies later stages of the software development process.

There has been limited research in using scenarios in the context of software development [9, 11, 19, 21]. As Table 1 illustrates, some of the findings from empirical research demonstrate that scenarios provide benefits for eliciting information requirements. Hsia et al. developed a process model of scenario analysis comprised of six sequential stages and the degree of user involvement in each stage. The stages include scenario elicitation, scenario formalization, scenario verification, scenario generation, prototype generation, and scenario validation (Figure 2). Scenario elicitation, as the name implies, is a stage of eliciting from appropriate individuals for gathering the required scenarios. In scenario formalization stage, each scenario is converted to construct a conceptual state that models system behavior from the perspective of an individual which leads to an abstract formal model. In scenario verification stage, the abstract formal model is verified for any inconsistencies, redundancies, and incompleteness. If any error is uncovered during this stage, the entire process is repeated. Scenario generation stage generates the scenarios that can be readily interpreted by developers and a prototype is developed during the prototype generation stage. Finally, in scenario validation, the prototype is used to validate the scenarios. If an invalid scenario is uncovered, the entire process is repeated.

3.1. Assumptions

Several assumptions are made in this scenario model. The first assumption is that the sequential model requires completion of the current stage before moving on to the next stage. While this may be true for some occasions, this is viewed as very limiting. For example, elicitation for the required scenarios are all conducted prior to scenario formalization stage. In an organizational setting, sometimes it is not possible to meet all the involved parties for gathering the required scenarios (as depicted in this project) before having to formalize them due to their availability and the project schedule.
The work must continue, for the time being, without including some input. It could be that an individual who is not available at the time for elicitation is in a unique department who is more than likely to provide a new scenario.

The second assumption is that scenario verification and validation stages are the only two stages where errors can be detected. In conjunction with this assumption is the third assumption that the analyst, not the developer, constructs the prototype and validates the scenarios. In a well designed software development process, whether a rapid prototyping or full software development application, it is not common to return to the end-user community before attempting to examine the code for any errors which the analyst is not necessarily qualified to do.

These assumptions do not necessarily reflect a software development effort in an organizational setting. Hence, a new model for conducting scenario analysis is presented based on a software prototyping effort in one organization. Prior to presenting a new model for scenario analysis, it may be appropriate to describe the role of knowledge within information requirements analysis phase of the software development process for the project under study.

3.2. Requirements Analysis

Traditionally, the information requirements have been gathered by means of analyzing the business processes of the organization and conducting interviews with and/or group sessions of the end-user community [6]. Unlike the traditional mode, there has been much attention given to the topic of knowledge acquisition recently for tapping into explicit and tacit knowledge of an individual, namely experts [3]. Knowledge acquisition is defined as “the transfer and transformation of problem-solving expertise from some knowledge source to a computer program” [3].

It is reasonable to believe that explicit knowledge primarily was gathered during the information requirements analysis phase under the traditional mode. There are several reasons underlying this statement: (1) traditional software development methods (e.g., waterfall, RAD, etc.) primarily may have involved the end-users of the system for eliciting requirements, who typically provided day-to-day operating procedures; (2) the infancy of technology and lack of guidance for software development process may have meant time spent on learning the technology and/or the software development process rather than devoting the same time to gathering information requirements; (3) lack of project management training and experience resulted in aggressive project schedules which may have sacrificed the quality of the information requirements; and (4) arrogance and lack of appreciation, at the time, for what is involved in designing an effective information systems may have lead to poor requirements. This is not by any means an exhaustive list, but it illustrates that the difficulty of effectively developing an information systems stems from lack of tacit and explicit knowledge as part of the information requirements analysis phase, particularly the former.

The question of “How can one go about eliciting tacit knowledge?” has been addressed to some degree by several researchers (e.g., [3]). Byrd et al., in particular, offers several knowledge acquisition techniques in their paper. However, the efficacy of these techniques rests largely on intuition and researchers’ experiences. Therefore, a prototyping project involving scenario analysis, structured interviews, and group observations is examined.

4. The Software Development Project: An Overview

A prototyping project involving the development of a system to document and evaluate current sales processes and identify areas of improvement with a focus on pricing strategy was undertaken in 1996 at one of the nation’s largest general contractor for expositions, trade shows, conventions and special events (the company wishes to remain anonymous). Lack of software development skills and human resources in-house forced the firm to seek outside expertise in carrying out the
objectives. A single project team was formed consisting of Chief Operating Officer, VP of information technology, one manager, two consultants at the project manager level, and two consultants at the analyst/developer level. Access to upper management, middle managers, and end-users of the system was provided, with high priority given to the development project due to its aggressive schedule and increased competition.

The success of the development project hinged on two factors - completeness of the system and timely completion. Both factors were considered equally important. Without the luxury of allocating sufficient time to gather the information requirements through comprehensive structured interviews and group sessions, scenario-based technique became an alternative to meeting this need. Scenario-based technique was chosen for the following reasons: (1) a managing consultant had experience with the technique; (2) the business process was not fully understood by any one individual; (3) it had the potential of capturing most, if not all, of the information requirements given the short time frame for the task; and (4) it had the potential of surfacing events that were considered “outliers.” The last point is an important one because identification of “outliers” was really the beginning point for acquiring tacit knowledge.

4.1. Eliciting for Knowledge

The firm under study used structured interviews, scenario, and prototype elicitation techniques in a software development effort. Participants included upper management executives, middle managers from various units (e.g., sales management, accounting, marketing, etc.), and end-users. In particular, organizational information requirements were gathered through upper management executives and middle managers and detailed information requirements for the application were gathered through middle managers and end-users [6]. The software development project experienced all four modes of the knowledge creation as suggested by Nonaka [14] - combination, internalization, externalization, and socialization.

4.1.1. Combination. Initially, structured interviews served to understand and document the existing business processes. This technique was very useful for obtaining explicit knowledge from explicit knowledge, i.e., combination mode of the knowledge creation. For employees of the firm, the existing business processes are the very processes under which they operate on a daily basis. This knowledge is more than likely documented and/or held by individuals whose activities depend on such processes. It is knowledge that can be transmitted through formal, systematic language at low cost.

Combination of structured interviews and scenario techniques was used to identify, through many iterations, the ranges of different types of business processes and the likely “what-if” scenarios for each type of business process. “What-if” scenarios are detailed step-by-step procedure for what an end-user would do to complete the sale under normal circumstances (rather than the exception). The scenarios were readily available to the consultants conducting the information requirements gathering since the end-users are intimately familiar with the process. The “usual” ways of conducting the business are explicit information held by each individual, and lend themselves to be transmitted without the involvement of tacit knowledge.

The “what-if” scenarios at this point focused on current business processes, and lead to the identification of goals for each scenario. The consultants, who were responsible for gathering the information requirements, were learning the industry, the business environment, the business processes, and the specific details of the day-to-day operations.

4.1.2. Internalization. From the perspective of those gathering the information requirements, a great deal of learning took place throughout the project and much of the knowledge was assimilated, digested, and “internalized.” The consultants were beginning to understand the industry and the business processes of the firm and, more importantly, the rationale for why the firm was doing what they were doing. The consultants were converting explicit knowledge into tacit knowledge, i.e., internalization mode of knowledge creation.

Through structured interviews, group sessions, and prototype sessions, the consultants were able to understand the business processes. During the structured interviews, various “work-around” techniques performed by the end-users were discussed to overcome sales terms considered exception to the norms. Reasons for handling a particular situation with certain “work-around” technique were discussed which enabled the consultants to better understand how the business was running in reality. The information requirements gatherers were beginning to “internalize” and assimilate the end-users’ explicit knowledge and converting it into their tacit knowledge.

During group discussions and prototype sessions, discussions centered around “unusual” scenarios, or sales that are exception-to-the-rule or “outliers,” surfaced even more tacit knowledge to the information requirements gatherers. For example, one end-user would process an
“outlier” sales one way for certain reason while another end-user would process the same “outlier” sales another way for a different reason. Since both ways are not incorrect, the opportunity for the consultants to learn was enormous. Each end-users’ explicit knowledge contributed to the consultants’ tacit knowledge. Continuous learning, and confirmation of their learning, allowed the consultants to build the tacit knowledge required to successfully develop an effective information systems.

4.1.3. Externalization. In the process of defining all possible “what-if” scenarios, explicit knowledge was obtained from tacit knowledge, i.e., externalization mode of the knowledge creation. The consultants during the interviews continually asked “why” to better understand the reasoning and logic behind why their business was being conducted the way they were. In other words, the consultants were tapping into the interviewee’s tacit knowledge. Not surprisingly, interviewee from the sales force frequently mentioned similar statement as follows:

“Although we are suppose to do (that) and (that) is how the business should be run, it doesn’t always work that way. So I have to do some juggling to close the deal.”

Knowledge of taking control and taking certain action under unique circumstances could be viewed as tacit knowledge. Using scenario analysis, structured interviews and prototype sessions, unique situations during a simulated sales process using a selected scenario were presented for inclusion into the deliverable system (as long as it was deemed “within” the project scope). The step-by-step procedure by which the end-users handled the unique situations were documented. The group approach (using the prototype) allowed the consultants to capture the commonality amongst the end-users, as well as the differences. Varied tacit knowledge was surfaced, as a result.

In other structured interview sessions with upper management executives and middle managers, each interviewee was given the opportunity to make improvements in the existing business processes by providing as many scenarios as possible five years into the future. Probable (predictable) and possible scenarios led to the identification of key drivers for changes in the business processes and the organization necessary to learn and remain competitive. Upper management executives and middle managers tacit knowledge were transformed into explicit knowledge.

4.1.4. Socialization. The process of converting tacit knowledge into tacit knowledge, i.e., socialization mode of knowledge creation, was primarily limited to end-user observation sessions, scenarios analysis and prototyping sessions. End-user observation sessions were activities whereby consultants watched the end-users in action. Value to individual observation sessions included those activities which were difficult to describe or document. As the saying goes “pictures speak more than thousand words,” the activities observed were events which would be difficult or very costly to document and transfer. In addition, the activities would have been difficult or very costly to interpret and understand even if those activities were somehow communicated or documented. The conversion of explicit knowledge to tacit knowledge during the observation became very evident given the amount of learning that has taken place, but the conversion of tacit knowledge to tacit knowledge was made possible through the experience of observing and listening to how end-users handle problems in action.

The consultants were also able to develop tacit knowledge through scenario formulation, development and refinement. They were continually reiterating the scenarios in various contexts to ensure that all information requirements were satisfied. New tacit knowledge were discovered by the information requirements gatherers almost every time they were engaged in a discussion with any employee of the firm. The same consultants benefited by designing and developing the prototype because they were embedding the scenarios, or requirements, into the system. Tacit knowledge held by the consultants were growing and continually being reinforced.

Prototyping sessions provided the medium for much of tacit knowledge conversion from both explicit and tacit knowledge. End-users were given the opportunity to test the system prior to its completion. There were namely two benefits to prototyping sessions for the firm under study: (1) an opportunity to make changes to the interface to obtain end-users’ buy-in and (2) an opportunity to validate the information requirements set forth in the scenarios. End-users validated the information requirements set forth in the scenarios and surfaced substantial number of issues during the prototyping sessions. First, many explicit knowledge were overlooked during scenario formulation, development and refinement. For example,

“When assigning Electrical Revenue, a salesperson needs to enter percentage of electrical revenue that is given to the local city which ranges from 5% up to 15% depending on the city and time of year.”

Some explicit knowledge were such a part of end-users daily lives that it was during development that several critical explicit knowledge were surfaced. Second, many issues considered to be tacit knowledge were being
surfaced. Namely, the issues revolved around the lack of ability to manipulate the data which can lead to no sales. The existing process of working with pencil and paper provided the flexibility for end-users to control some aspects of the sales. However, the system would limit such flexibility. Detailed discussions of what really happens in the field allowed the end-users to surface their tacit knowledge. On the other hand, the information requirements gatherer (who were also developers) were understanding the process more than any one individual in the organization. The consultants were creating and acquiring tacit knowledge.

In summary, the consultants experienced continuous interchange between explicit and tacit knowledge throughout the information requirements analysis phase (as well as the programming phase). The knowledge acquired and created was made possible through usage of multiple elicitation techniques. Specifically, scenario technique proved to be an important asset for capturing the information requirements necessary for developing an effective information system. A scenario analysis model is proposed below based on the project just described.

4.2. Scenario Analysis

While the scenario model provided by Hsia et al. was applied successfully to a PBX system, this project does not appear to align with their model primarily due to violations of their assumptions. In addition, there were more feedback loops in this project than suggested by Hsia et al.’s scenario model. Another reason is that the process of gathering the scenarios differed from that of Hsia et al. This section is divided into three areas: (1) Functional Process of Upper Management depicting the development of scenarios from the upper management’s perspective; (2) Functional Process of End-Users depicting the development of new scenarios and refinement of scenarios identified by the end-users; and (3) Technical Process depicting the refinement and programming of scenarios identified by all participants.

4.2.1. The Functional Process of Upper Management. The consultants conducted several structured interviews with upper management (who were not the end-users) and obtained several real-life scenarios of their business. The structured interviews with the upper management were appropriate given that the end-users may not necessarily be fully aware of all the intricacies involved in the business. Furthermore, these examples can serve as checks and balances to ensure that the upper management’s perception of the business aligns with the end-users’ perception of the business while addressing the need raised by Davis [6].

The structured interviews were held on a one-on-one basis due to difficulty of having to form a meeting where everyone can be present. This forced the consultants to document the scenarios from the first interview and identify key drivers to arrive at a more general scenarios, which resembles the scenario formalization stage. The scenarios were then presented during the second structured interview for verification, which is equivalent to scenario verification stage. In addition, the interviewees were given the opportunity to identify other scenarios that may not already be represented in the scenarios presented. If there were no scenarios provided or no modifications suggested, the same set of scenarios were presented to the third interviewee. If there were additional scenarios and/or modifications suggested, the scenarios were translated into more generalized scenarios and/or modifications made prior to conducting the third structured interview. The process repeated itself until all the structured interviews were completed. This entire process ensured that all the possible scenarios were represented, at least from the upper management’s perspective, and represents scenario elicitation, formalization, and verification as depicted in the Functional Process of Upper Management in Figure 3.

4.2.2. The Functional Process of End Users. An initial set of scenarios were now documented representing the upper management’s perspective. The next step was to document all possible scenarios from the end-users’ perspective. Some structured interviews and group sessions were held with designated end-users. During the structured interviews and group sessions, each scenario obtained from the upper management was presented one at a time. End-users worked with each scenario to refine, add, or remove anything they felt was inappropriate. It became very evident that the end-users were intimately familiar with the details of day-to-day activities. The group sessions provided two additional scenarios overlooked by the upper management (reflected in scenario elicitation and formalization in the Functional Process of End Users in Figure 3). The entire process confirmed scenarios provided by the upper management as well as added additional ones. The end-users scenario process includes scenario elicitation, formalization, and verification as depicted in the Functional Process of End Users in Figure 3.

4.2.3. The Technical Process. A complete set of scenarios were documented and abstract formal models were generated for development. Using the generated scenarios, the development team developed a prototype of the expected system. The developers tested the prototype
using the scenarios to ensure it was working as expected. After detecting an error, developers relied on the generated scenarios for guidance. If the problem could not be resolved, the development team documented them and resolved them with the end-users during the weekly meeting (depicted by arrow 2 in Figure 3). The weekly meeting was set up to discuss whatever was necessary to ensure that the project would meet its goals and to communicate to the end-users with its progress.

After correcting for any errors or changes to modification requests, a formal session was held to demonstrate the prototype to the end-users. If errors were detected, the developers initially checked the prototype for coding or logic errors. Otherwise, the end-users either accepted the prototype or made cosmetic change requests. There were two occasions during the development of the prototype where end-users raised important issues. The first issue dealt with providing an additional screen displaying a report that was normally just printed. The issue implied that there was going to be an increase in the scope of the project. The issue was propagated to the upper management for resolution, depicted by arrow 1 in Figure 3. The second issue dealt with adding functionality that was not part of the scenarios. This functionality was part of the Request-for-Proposal and was overlooked. The project manager quickly made a decision to incorporate the functionality, at which point the end-users provided the detailed specifications necessary to program the added functionality. Fortunately, the added functionality did not change any part of the scenarios developed. Otherwise, it would be necessary to revisit each and every scenario for inspection.

The relationship of scenario generation, prototype generation, and scenario validations is depicted in the Technical Process in Figure 3. Scenario analysis proved to be very effective for this project, but appear to be very different from the model proposed by Hsia et al. One major difference between this project and Hsia et al.’s model is the varying level of analysis [6, 18]. Hsia et al. considers the end-user community to be the only source for input, whereas this project uses the upper management (who were not the end-users), the middle managers, and the end-user community. For this reason, the model proposed for conducting scenario analysis is depicted in Figure 3.

5. Discussion and Future Research

Information requirements phase is an important part of the software development process. Incorrect, incomplete, or absence of information requirements can be detrimental to the success of the software development project. This paper illustrated the value of combining several elicitation techniques for gathering information requirements by describing one firm who exercised the use of structured interviews, scenario analysis, group discussion and observation sessions, and prototype techniques. In particular, scenario technique was described for surfacing both explicit and tacit knowledge held by the individuals of the firm. The software
development project described in this paper adds to the existing literature in three ways: (1) the project makes use of more than one elicitation technique; (2) a model of scenario technique is suggested; and (3) the project is based on an empirical evidence of one firm. Despite interesting results, this paper only demonstrated what only one firm experienced. In agreement with many researchers, there needs to be more empirical studies to better understand the uses of various elicitation techniques for gathering information requirements.

For researchers, this paper extends the existing literature by providing empirical evidence of multiple elicitation techniques used by one firm as suggested by Byrd et al. [3]. Furthermore, a scenario-based model is proposed for gathering information requirements which is subject to further empirical testing. For practitioners, an alternative to traditional information requirements techniques is offered as a way to enhance the development effort, along with a scenario analysis model as a guidance.

Observations made from the software development project described in this paper offered several interesting future research topics. In particular, explicit and tacit knowledge and scenario-based technique research topics will be addressed because they seem to be the most interesting.

With respect to explicit and tacit knowledge creation and conversion, Nonaka [14] offers his Spiral of Knowledge Creation model and places importance on the interchange between explicit and tacit knowledge through internalization and externalization. This was observed during the information requirements gathering phase of this project and it appeared that all four modes of knowledge creation was taking place simultaneously. This observation is consistent with Nonaka’s claim that while each of the four modes of knowledge conversion can create knowledge independently, the more valuable knowledge stems from a dynamic interaction between the different modes of knowledge creation. Theoretically, all explicit and tacit knowledge should unveil through the dynamic interaction between the different modes of knowledge creation. However, the firm under study revealed that not all explicit and tacit knowledge can be transformed (e.g., scope creep during programming phase of the software development process). It is not clear why this is the case despite the presence of all four modes of knowledge creation.

There was a recognition in this project that not all tacit knowledge comes from the experts. Traditionally, it is believed that tacit knowledge is primarily held by the experts. During the information gathering process, the development of tacit knowledge was not obtained by the experts, but rather multiple individuals. Individuals collectively provided most of the tacit knowledge required to develop the information system. This challenges the assumption that tacit knowledge is held by the experts. Could it be that tacit knowledge held by multiple individuals outweigh the tacit knowledge held by one or more experts? Needless to say, this is not an easy question to address, however, it raises an issue fundamental to the topic of tacit knowledge and expert systems.

A scenario analysis model was proposed as a technique for gathering information requirements. It appears that it would be difficult to obtain high-quality information requirements as a standalone technique. Furthermore, it appears that interviews are indeed necessary for gathering complete information requirements, as demonstrated in this study. A research could examine which other elicitation techniques complement and support the scenario technique.

Finally, during a software development process, the analysts conducting the information requirements are usually different resources than the developers. However, the analysts in the project described were also developers for the same project. It would be interesting to examine what changes would be necessary to the scenario analysis model to account for such difference. More importantly, how much explicit and tacit knowledge would be lost during the transfer? And, how much loss would there be in the information system being developed in terms of completeness?

6. Conclusion

This paper examined the role of scenario-based technique in information requirements analysis phase of the software development process. Specifically, a prototyping effort in a for-profit organization was examined who developed an effective information system. Their experience with scenario-based technique, in conjunction with the well known structured interviews and group session techniques, proved to be invaluable in determining information requirements. As a result of the thoroughness and completeness of the requirements, the design and the development of the prototype met its goals by delivering the completed system on time.

Scenario-based technique should not be taken as a solution or a better alternative to the traditional elicitation techniques, but rather as a technique which may be preferred over another technique or as a technique which may supplement another technique. More empirical research is necessary to further understand the process of gathering information requirements, and this paper provides one view of how
one firm used various elicitation techniques to develop an effective information system.

References


