8-2013

Improving Requirements Generation Thoroughness in User-Centered Workshops: The Role of Prompting and Shared User Stories

Aaron Read

University of Nebraska at Omaha

Follow this and additional works at: https://digitalcommons.unomaha.edu/studentwork

Part of the Computer Sciences Commons

Please take our feedback survey at: https://unomaha.az1.qualtrics.com/jfe/form/SV_8cchtFmpDyGfBLE

Recommended Citation


This Thesis is brought to you for free and open access by DigitalCommons@UNO. It has been accepted for inclusion in Student Work by an authorized administrator of DigitalCommons@UNO. For more information, please contact unodigitalcommons@unomaha.edu.
Improving Requirements Generation Thoroughness in User-Centered Workshops: The Role of Prompting and Shared User Stories

By

Aaron Read

A DISSERTATION

Presented to the Faculty of

the Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Doctor of Philosophy

Major: Information Technology

Under the Supervision of Dr. Gert-Jan de Vreede

Omaha, Nebraska

August, 2013

Supervisory Committee:

Robert O. Briggs
Doug Derrick
Ann Fruhling
Roni Reiter-Palmon
Harvey Siy
The rise of stakeholder centered software development has led to organizations engaging users early in the development process to help define system requirements. To facilitate user involvement in the requirements elicitation process, companies can use Group Support Systems (GSS) to conduct requirements elicitation workshops. The effectiveness of these workshops for generating a valuable set of requirements for system developers has been previously demonstrated. However, a more representative measure of progress towards a system that will meet users’ needs—the completeness of the requirements generated by such groups has not been explored. We explore two process design considerations for increasing the completeness of requirements generated by these users: increased sharing of user stories (individual electronic brainstorming groups vs. shared user stories electronic brainstorming groups), and the use of reflective inducement prompts (unprompted vs. prompted groups). Using the Search for Ideas in Active Memory model, we predict that prompted electronic brainstorming groups will outperform any other group, including prompted, shared user stories groups at generating a more thorough set of requirements. To test the hypotheses an experiment with 56 groups consisting of 197 users was conducted. The users were asked to generate requirements for a fictitious online textbook exchange website. All hypotheses received support. The study has implications for GSS-Supported workshop design and for future research on collaborative performance in requirements elicitation.
Acknowledgements

Many thanks to Ron and Melanie, my parents, for both pushing me to strive for excellence and for providing encouragement. A great thank you to my advisors Bob Briggs and GJ de Vreede—for both pushing me and for believing in me as an academic thinker. My time at UNO has been transformational. Thanks to my roommates and other special friends.

Also thanks to the student coders who put lots of city mileage on their brains to read thousands of responses and place them in dozens of categories.

Thanks to my committee for providing me with useful feedback.

Thanks to Doug Dean who started me on the path to research almost 10 years ago.
## Contents

Acknowledgements .............................................................................................................. 1

1. Introduction .................................................................................................................. 1
   1.1 Software Requirements ......................................................................................... 1
   1.2 Importance of Software Requirements ............................................................. 1
   1.3 Requirements Completeness .............................................................................. 2
   1.4 Requirements Completeness Challenges ......................................................... 4
   1.5 Wide User Audience System Stakeholder Involvement ........................................ 5
   1.6 Improving Requirement Generation Thoroughness in Requirements Development Workshops ......................................................................................................................... 7

2. Requirements Determination ........................................................................................ 9
   2.1 Introduction to the Field of Requirements Determination .................................. 9
   2.2 The Role of Requirements Determination in Software Development ................. 10
   2.3 The Requirements Determination Process ....................................................... 12
   2.4 Requirements Determination Techniques ....................................................... 15
   2.5 The Value of User Group Requirements Elicitation Workshops in Requirements Development ......................................................................................................................... 21

3. Increasing Requirements thoroughness in User Centered Requirements Elicitation Workshops ......................................................................................................................... 23
   3.1 Previous Research on User-Centered Requirements Elicitation Workshops ........ 23
   3.2 Previous Research in Requirements Elicitation Techniques ............................... 25
   3.3 Overview of Human Cognition in Requirements Elicitation ............................. 25
   3.4 Increasing Requirements Thoroughness ......................................................... 32
   3.5 Group Requirements Elicitation ....................................................................... 33

4. Presentation of the Research Model ............................................................................ 38

5. Method ......................................................................................................................... 44
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Participants</td>
<td>44</td>
</tr>
<tr>
<td>5.2</td>
<td>Procedure and Tasks</td>
<td>45</td>
</tr>
<tr>
<td>5.3</td>
<td>Measures</td>
<td>49</td>
</tr>
<tr>
<td>6.</td>
<td>Data Analysis and Results</td>
<td>50</td>
</tr>
<tr>
<td>6.1</td>
<td>Hypotheses Tests</td>
<td>50</td>
</tr>
<tr>
<td>6.2</td>
<td>Comparison of Raw User Stories Generated</td>
<td>55</td>
</tr>
<tr>
<td>7.</td>
<td>Discussion</td>
<td>56</td>
</tr>
<tr>
<td>7.1</td>
<td>Limitations and Future Research</td>
<td>61</td>
</tr>
<tr>
<td>7.2</td>
<td>Practical Implications</td>
<td>68</td>
</tr>
<tr>
<td>7.3</td>
<td>Conclusions</td>
<td>68</td>
</tr>
<tr>
<td>8.</td>
<td>References</td>
<td>69</td>
</tr>
<tr>
<td>Appendix A</td>
<td>BookExchange</td>
<td>84</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Requirements</td>
<td>85</td>
</tr>
<tr>
<td>Appendix C</td>
<td>Coding Rules</td>
<td>92</td>
</tr>
<tr>
<td>Appendix D</td>
<td>Informed Consent</td>
<td>93</td>
</tr>
<tr>
<td>Appendix E</td>
<td>Exploration of the Effectiveness of Individual Prompts</td>
<td>96</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Software Requirements

Software is only valuable to the extent that it meets the needs of a variety of stakeholders who will benefit from, use, design or otherwise succeed through its completion (Boehm, 2003). To effectively meet stakeholder needs, the software must also be internally consistent and reflect potential technical, process, organizational, social, legal, and political constraints (Boehm et al., 1994, Baskerville et al., 2004, Hopkins and Jenkins, 2008). These constraints, along with the goals of stakeholders are discovered as part of an intensive knowledge gathering process (Hickey and Davis, 2004). When stakeholders have adequately discovered and reached agreement about the needs and constraints, which will be reflected in the design of the software, a set of requirements is created. Requirements are statements of agreement about constraints on the behavior of the software and the goals which the design of the software will fulfill (Sommerville, 2007). They are statements which can be ultimately verified as the developed software is tested and experienced by stakeholders (Davis, 1993).

1.2 Importance of Software Requirements

Requirements play a foundational role in the development of software. Requirements are the direct input for software design and subsequent code development. Poorly developed or incomplete requirements remain one of the major sources of software development project failure (Hofmann and Lehner, 2001, Baccarini et al., 2004). If errors introduced in the requirements are not caught, they become more costly to fix
once the requirements have been turned into software code (Boehm, 1991, Lutz and Mikulski, 2003). It is largely due to the difficulty of developing complex software requirements that computer software costs remain high while hardware costs have decreased drastically (Research Triangle Institute, 2002, Brooks, 1987).

1.3 Requirements Completeness

One of the critical quality goals of requirements is that they do not contain omissions or reflect a lack of understanding which would introduce undesired characteristics into the system. Completeness of requirements is difficult to define and difficult to detect (Davis, 1993). Boehm (Boehm, 1989) states that requirements have three fundamental characteristics to be complete. When examining a requirements document, one should find that:

(1) No information is left unstated or “to be determined”
(2) The information does not contain any undefined objects or entities
(3) No information is missing from the document.

However, it is difficult to know what constitutes “missing information.” Boehm has stated that the requirements for a project may not be completely known until the project is complete (Boehm, 2000), because much of the requirements will be discovered through learning as the design of the software matures. In other words, requirements become more complete as they reflect increased knowledge (Hickey and Davis, 2004).

The knowledge which increases requirements completeness originates from two sources: the knowledge success critical stakeholders have of their own and other stakeholders’ goals, and the knowledge they have of any constraints on the proposed system. The goals of success critical stakeholders are of prime importance. The
requirements process should proceed with the consent of success critical stakeholders, or those individuals whose interests must be accommodated for a project to succeed (Sharp et al., 1999), or otherwise stakeholders may resist by refusing to cooperate (Jiang et al., 2000). The involvement of stakeholders such as product owners and customers also assures that the development team is developing the functionality that is actually needed. Involving users, for example, encourages a more positive attitude towards the system under development (He and Wei, 2009). Dissent and subsequent clarification and negotiation play a major role in the requirements development process as teams of stakeholders work towards shared understanding and shared agreement about a proposed set of requirements (Chakraborty et al., 2010, Boehm et al., 2001).

Success critical stakeholders must not only consider the functionality they desire for the system, but also be aware of the realities surrounding the system which will determine its success. For example, a system must be compatible with other systems which form part of its operational environment (Kaefer and Bendoly, 2004). Stakeholders must consider resources available to develop the system (Thong, 2001). There are also a myriad of invisible or non-functional considerations such as maintainability, reliability, and testability which can also impact the success of the system (Chung and Prado Leite, 2009). Knowledge of these constraints is critical to the success of the system.

Requirements are complete to the extent that they reflect both knowledge of success critical goals as well as constraints on the system. In other words, requirements are complete to the extent that a) the requirements accommodate the goals and constraints that all success-critical stakeholders agree are relevant for the system; b) all success critical stakeholder agree as to which goals and constraints the system should not
accommodate; and c) there are no undiscovered goals or constraints. “Success-critical stakeholder” will be abbreviated to “stakeholder” for the remainder of this document.

1.4 Requirements Completeness Challenges

There are several challenges inherent to the requirements development process. Because stakeholders of various backgrounds and expertise are often included in the requirements development process to assure that the software will meet their goals, cognitive and social limitations of humans must be considered as an inherent challenge in the requirements development process (Orlikowski and Gash, 1994).

When non-technical stakeholders are involved in the software development process, it can difficult to obtain requirements that will assure that the software successfully meets their needs for several reasons. Non-technical stakeholders will tend to understand the system from a functional perspective while technical stakeholders such as developers will understand the system in terms of its technical characteristics (Orlikowski and Gash, 1994). Requirements for software development are also difficult to express. Requirements do not exist in the minds of stakeholders as a suitably structured set of statements about the desired behavior of the software (LaFrance, 1992). Rather, they exist as tacitly held in a set of assumptions and knowledge which the stakeholder holds unconsciously to make sense of and complete tasks in their work environment (Buchan and Ekadharmawan, 2009). These assumptions and this knowledge has become unconscious to the stakeholder as tasks have become automated through expertise (Sweller et al., 1998). Stakeholders often leave requirements partially unexpressed or ambiguous, leaving the development team to make assumptions about the software (Globerson, 1997) or to develop something that ultimately does not satisfy the
stakeholders. Customers and product owners who participate in the requirements development process are, therefore, often unable to express their needs easily or adequately (Buchan and Ekadharmawan, 2009).

1.5 Wide User Audience System Stakeholder Involvement

In spite of the challenges presented by involving users in the requirements development process, there has been an increasing interest in involving a wide audience of users in development projects where many users with diverse needs are the primary beneficiaries of the software (Bragge and Merisalo-Rantanen, 2009, Tuunanen, 2003). This is especially true given the increasing success of internet-based systems where users not only consume but produce information (Iivari and Iivari, 2010). At the same time, there is also potential for users to contribute to the quality of the requirements of the system. Users can be good innovators when developing new services or products (Magnusson et al., 2003, Matthing et al., 2006). User involvement broadens the empirical scope of system design (Carroll, 1996). User involvement encourages developers to take users’ needs seriously (Kuhn, 2000).

There are several successful instances of groups eliciting requirements indirectly in requirements development workshops or focus groups (Hengst and Vreede, 2004, Bragge et al., 2005, Tuunanen, 2003, Peffers et al., 2003). While the contributions from these groups are not directly incorporated as requirements in the design process, they provide inspiration for user functionality, and confirm the importance of envisioned functionality to targeted end users. Bragge and colleagues (Bragge and Merisalo-Rantanen, 2009) demonstrated that users can generate a substantial amount of
requirements for system designers. While the developers had thought of many of the requirements, the user-generated contributions provided system developers with a sense of the user’s priorities of requirements, and left users and developers satisfied with the process. This gives the users a sense of involvement and being properly represented.

Users are a special group of stakeholders with special considerations. The consideration of the needs of users as stakeholders has been increasingly advocated in recent years (Hofmann and Lehner, 2001), especially for internet-based systems which have a wide range of users (Bragge and Merisalo-Rantanen, 2009, Tuunanen, 2003), and where the contributions of user content play a large role in the value provided by the system (Iivari and Iivari, 2010). The involvement of the user can take on many forms, varying in involvement from seeking to understand an envisioned, fictitious user (Norman, 1986), to providing feedback for the system (Bragge and Merisalo-Rantanen, 2009), to actually providing direct input for the design (Iivari and Iivari, 2010). Direct user involvement, especially in a new product development environment can be difficult (Iivari, 2006), because it is difficult to contact or identify potential users. Often representative users are selected to stand in for a larger group of users.

Groups of users can participate simultaneously and efficiently in development workshops (Bragge and Merisalo-Rantanen, 2009). However, understanding of how well the users in such workshops provided actual system worthy data is left largely untouched. He et al.’s meta-analysis of the benefits of user participation (He and Wei, 2009) revealed a contrast between the more influential positive effects of user involvement on user attitudes and behavior with respect to the system and the weaker effects of user involvement on development productivity and quality. The lack of influence of users on
the productivity development of the system is largely attributed to the inability of users to provide helpful recommendations for the system, especially when users participate without training or guidance (Peffers et al., 2003). Some researchers suggest that user involvement could improve if cognitive limitations are addressed. For example, the appropriate use of prompts and questions has been shown to improve the quality of requirements in interviews between analysts and stakeholders or other user representatives by facilitating the ability of users to recall system-related knowledge (Browne and Rogich, 2001, Moody et al., 1998, Peffers et al., 2003). While the benefit of these prompts has been demonstrated in one-on-one interviews with stakeholders, including users, to the best of my knowledge, the benefit of cognitive prompts for novice users who would likely participate in focus groups or workshops has not been demonstrated yet.

1.6 Improving Requirement Generation Thoroughness in Requirements Development Workshops

The current study investigates the impact of cognitive prompts, or one sentence instructions given to individuals to focus their thinking during brainstorming, on the ability of users to generate more a thorough set of requirements while participating in brainstorming sessions of Group Support System (GSS) supported requirements development workshops. Requirements thoroughness is defined as the extent to which requirements reflect the knowledge of stakeholders and their goals without knowledge of their The ultimate goal of this research is to improve requirements completeness. However, it is difficult to improve requirements completeness only within a brainstorming session of a brainstorming session. Requirements must be validated and
prioritized amongst stakeholders through careful communication (Coughlan and Macredie, 2002). This cannot occur within a requirements workshop. We can, however, test the extent to which individuals have been thorough in their thinking about the details of requirements, which could then be presented later for validation and prioritization amongst stakeholders.

Because individuals in these workshops are also exposed to the ideas of others, the effects of the user stories generated by others on requirements thoroughness will also be assessed. Group brainstorming research has for decades explored the extent to which a collaborating group is able to outperform a group of individuals performing electronic brainstorming at producing novel or creative ideas. This research has found that the difference in the number of quality ideas generated between groups and individuals can be accounted for largely in terms of the costs and benefits of being exposed to external stimuli for knowledge activation (Stroebe et al., 2010). This research has been fruitful, leading to theories of group ideation such as Bounded Ideation Theory (Briggs and Reinig, 2010) and the Search for Ideas in Active Memory (SIAM) model (Nijstad and Stroebe, 2006) which have been used to explain the effects of external stimuli on an individual’s idea generation performance, including both the ideas of others (Nijstad et al., 2003) as well as prompts administered by facilitators (Santanen et al., 2004) to collaborating groups.

The current study will attempt to demonstrate that knowledge activation is at the core of both successful idea generation and the recall of knowledge necessary for the requirements development brainstorming task. The current study will integrate group brainstorming research into a study of group requirements elicitation. The effects of
collaboration and prompting on the completeness of requirements generated in GSS supported workshop settings are investigated in a lab experiment, where subjects generate requirements for an online book exchange system. To establish the contribution of this study within the context of the requirements development process, the next chapter discusses the requirements development field and process. Chapter 3 discusses previous research on requirements elicitation and how the current study builds on this knowledge and integrates this knowledge into the SIAM model and surrounding research on group productivity. This chapter concludes with the presentation of the hypotheses to be tested. Chapter 4 describes the method used to test the hypotheses. Chapter 5 presents the results of the study. Chapter 6 concludes the dissertation with a discussion of overall results, including limitations of the study, directions for future research, and conclusions.

2. Requirements Determination

2.1 Introduction to the Field of Requirements Determination

The study of requirements development constitutes an entire branch of software engineering, largely due to the complexity of the task and the numerous techniques and methods developed to accomplish it (Mathiassen et al., 2007). Within requirements development, requirements elicitation is only one of several critical activities. Eliciting requirements from groups of users—is only one possible technique for eliciting requirements, eliciting requirements being only one of several critical activities in the requirements development process.

The value of the requirements development process becomes clearer when its role within the development of software is understood. The requirements development
process also constrains the manner of software development. The quality of the code developed by the software is largely determined by the quality of the requirements. The role of the requirements development process in the overall software development process is therefore first visited in section 2.2.

To understand the value of a requirements development activity, it is necessary to be aware of the specific requirements-related problem the activity is trying to solve. Requirements evolve from their beginning state to more mutually understood specifications through a series of generally recognized steps which delineate a problem the requirements development team is trying to solve. The requirements development process is therefore reviewed in section 2.3.

Finally, the value of eliciting requirements from groups of users can be better understood when it can be compared to similar techniques or prescribed ways of performing requirements development activities. Different types of requirements development techniques are described in section 2.4.

Finally, in section 2.5, group requirements elicitation is described as an activity in the requirements development process and as a requirements development technique. The expected contribution to requirements thoroughness for this technique is also described.

**2.2 The Role of Requirements Determination in Software Development**

Requirements are a critical deliverable in the software development process. Requirements represent explicitly stated decisions about what the software should do. They should describe what the system does but not how it is done (Siddiqi, 1994). Requirements are not only used for the initial design of software, but are also relied upon
during later development tasks such as testing and maintenance (See Figure 1 below for an overview of the software development process). Software is not only tested to assure that glitches unintended by the programmer are caught, but also to assure that the software is what the user or customer envisioned. In the maintenance phase, where software is updated to meet the changing needs of an organization, it is useful to know when a change may potentially conflict with an already existing functionality of the software, which is stated in the requirements. Quality traits of requirements such as those set forth by IEEE (IEEE, 1998)—correctness, a lack of ambiguity, completeness, consistency, verifiability, modifiability, and traceability (Gotel and Finkelstein, 1994), and prioritization become more apparent as the demands of these later tasks are considered. In iterative software development cycles, requirements may not reach such a high level of quality in documented form, but are confirmed through face-to-face interaction with the customer (Cohn, 2004), sometimes while discussing already built software (Maiden and Rugg, 1996).

![Figure 1: Requirements Determination and the Software Development Process](image-url)
2.3 The Requirements Determination Process

Requirements are seldom developed in one sitting; they evolve as stakeholders develop a shared, complete, and specific understanding of stakeholders needs and constraints on the system (Pohl, 1994). While the requirements development process does involve a continuous gathering and refinement of knowledge (Hickey and Davis, 2004), there are distinct “states” within the process where a specific problem is being treated (Chakraborty et al., 2010). Understanding stakeholder’s needs, gathering information, negotiating priorities between stakeholders, analyzing and organizing requirements, and specifying requirements are all different areas of focus within the requirements development process (Sommerville, 2007, Browne and Ramesh, 2002, Byrd et al., 1992). Each of these activities contributes to the completeness of the requirements in different ways. For example, during a requirements negotiation meeting, the focus of the meeting is more about establishing agreement about the high level goals of the system than about assuring that the requirements are highly detailed.

The steps listed here represent the most commonly cited sequence of activities, but do not necessarily represent the way the requirements development process must unfold. In addition to being useful guidelines, they also demonstrate the evolution of the completeness of requirements as the goals of users and constraints on the system are progressively better understood.

1. Requirements Scoping

Chakraborty and colleagues (2010) describe a stage of requirements determination where the problem is defined and scope of the requirements is first determined. It is similar to Nonaka’s concept of originating Ba (Nonaka et al., 2000)...
where the shared context necessary for further knowledge exchange is established, but deeper mental models of the operation of the system have yet to be defined. A key aspect of this stage is the establishment of a mutually felt sense of the urgency of the core issues of the system. For example, user representatives may communicate the need to assure that the system implementation avoids the hiccups of the previous system, or the need for the system to comply with external regulations. It is during this stage that the social context is set and that trust is established between team members. It sets the stage for the more in-depth, tacit knowledge of the requirements to be shared later on. From a practical point of view, one key outcome of this stage is that the group is able to feel confident about the “go-no-go” decision for the project.

2. Requirements Negotiation

When there may be differences in opinion amongst stakeholders about which requirements should actually be included in the design of the system, and which are of greater priority, stakeholders meet to negotiate the requirements. It is during this stage that stakeholders agree which needs will be met by the system, which will not be met, as well as which constraints the system will address. Compromises may result in new requirements. An example of a method that has addressed this is EasyWinWin, developed by Boehm and colleagues (Boehm et al., 2001). In this method, stakeholders use a Group Support Systems tool to create a prioritized list of mutually beneficial requirements and discover the reasons for conflicting views on requirements. In the requirements negotiation activity, stakeholders focus on establishing knowledge of stakeholder goals and agreement. Often, at the same time, the stakeholders develop mutual knowledge of
high level constraints on the system as well, but they will not explore the implications of these constraints in great detail.

3. Requirements Elicitation

The requirements elicitation task concerns the development of understanding of stakeholder’s needs in greater depth. In-depth requirements elicitation activities usually follow the negotiation of high level requirements, and certainly follow the initial scoping of the project. Generally, requirements are elicited by analysts or developers from the stakeholders of the system. An analyst may make some assumptions and extrapolations based on his or her expertise. The stakeholder is probed both to encourage the revealing of more requirements (Pitts and Browne, 2007), as well as to allow the analyst to fill in gaps. The analyst may use a representation of his or her understanding of the requirements to aid reasoning about the requirements. This stage of the requirements elicitation process ends when there is an agreement between the stakeholder and the analyst about the content and meaning of the requirements, which can then become specifications.

Much of the empirical research of this stage of the requirements determination process focuses on the initial gathering of requirements before the analyst makes any attempt to reformulate, represent, and verify his or her understanding of the requirements. Users’ understanding of their software needs are often difficult to verbalize. Interviews and other techniques are used prompt stakeholders to explain their needs in terms that can be understood and verified by the developers. Depending on the maturity of the understanding of the needs to be fulfilled by the system, this complete set of requirements should include an understanding of users goals and existing processes, tasks, and
4. Requirements Analysis and Specification

In the specification stage, ideally, all tacit understanding has becomes an explicit commitment through documentation and conversation. Enough commitment and shared understanding has been reached in previous stages that details of the requirements can be filled in and analyzed for quality (completeness, consistency, etc.). In a waterfall approach to software development, a requirements specification document is the final delivery of this stage and of the requirements development process. In Agile software development methods, the details of agreements from conversations surrounding user stories are captured as acceptance tests (tests which are ran to verify that functionality worked correctly). It is during the specification stage that the IEEE qualities of requirements (e.g. correctness, a lack of ambiguity, completeness, consistency, etc.) become a focus of the requirements development team.

2.4 Requirements Determination Techniques

Research in the requirements determination field tend to be solution centric--evaluating, proposing, and validating implementable solutions or techniques for solving specific requirements development activities (Wieringa et al., 2006). The solution can be a (1) technique--a prescribed way to perform a certain activity, (2) a tool--an artifact (such as a software program or notation method) used to facilitate or automate a specific activity, or (3) a method—a prescribed way of using a collection of interrelated
techniques and tools (Nuseibeh and Easterbrook, 2000). Figure 2 illustrates the relationship between methods, techniques, and tools. In the current study, a requirements determination technique is proposed and evaluated.
To complete the various steps of the requirements determination process, there are dozens of techniques to choose from (Mathiassen et al., 2007). There has recently been an increased effort to identify the “why” or underlying rationale of a requirements determination technique. Such knowledge enables the selection (Hickey and Davis, 2004), and tailoring (Becker et al., 2007) of requirements determination methods, tools, and techniques given certain contextual factors.

Much of this research has been spurred on by the Agile movement (Boehm and Turner, 2003, Conboy, 2009) and other recent developments in the software industry which call for a need to present the knowledge of requirements in new contexts. For example, Agile software development methods are designed to enable a software team to respond rapidly and effectively to changing requirements. Previous methods advocated rigorous analysis of requirements which were documented and communicated to developers using formal notations while Agile methods advocate for minimizing documentation and formal analysis efforts (Turk and France. R, 2005). Several researchers have noted, however, that either approach need not be used exclusively, and that a blend of approaches is possible and desirable (Boehm and Turner, 2003, Mohan et al., 2010). Several theoretical frameworks have been proposed to help define the “spirit” or underlying rationale of Agile methods, so that a blend of methods can be assessed for its agility (Mohan et al., 2010) or an individual technique can be assessed (Conboy, 2009). For example, Conboy (Conboy, 2009, p. 340) defined the Agility of a technique as “the continual readiness of an Information Systems Development method to rapidly or
inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment.”

With such a definition, a researcher or practitioner can assess the extent to which a technique or set of techniques is Agile and should therefore be used when requirements undergo comparatively higher rates of change.

Changing requirements is not the only challenge faced by software development teams. Other factors, such as the ability to develop an architecture early, suggest the need for completeness in agility (Beck, 2005, Coplien and Bjørnvig, 2011), for example. Several researchers have proposed that the risks threatening project failure should ultimately determine the behavior of software development teams (Boehm and Turner, 2003, Mathiassen et al., 2007). As part of an effort to classify the dozens of requirements determination techniques presented in the literature, Mathiassen and colleagues (Mathiassen et al., 2007), present a risk-based classification scheme. In addition to risks associated with requirements change, they consider risks that arise from difficulties identifying requirements, and risks that arise from the complexity of requirements. Mathiassen and colleagues also present classifications for the underlying rationale of techniques. These underlying rationales explain how techniques respond to risks. When we consider these classifications of techniques, and the underlying rationales used to define them, we can understand the “why” of a requirements technique from a much broader perspective than the question of Agility. These classifications are discussed below.

1. **Requirements Discovery Techniques**
Requirements discovery relies heavily on stakeholder involvement. Stakeholders can be involved more extensively and in more passive or more active roles at different steps of the requirements determination process in order to aid in the identification of requirements (Iivari and Iivari, 2010). They play the most crucial role in requirements discovery, where user needs are surfaced and clarified. Discovery methods focus on overcoming the social, communication, and cognitive barriers to identifying requirements. They primarily reduce risks associated with correctly identifying requirements. In the waterfall method, stakeholders are typically involved through interviews, focus groups, and workshops at the beginning of the project. Usually a systems analyst takes charge of the requirements gathering process and is ultimately responsible for assuring the completeness and correctness of requirements.

Users are a special group of stakeholders which are involved both for an assessment of their needs, and as a means of improving attitudinal acceptance of the system (He and King, 2008). Users are a group of stakeholders with special considerations. The consideration of the needs of users as stakeholders has been increasingly advocated in recent years (Hofmann and Lehner, 2001), especially for internet-based systems as of a wide range of users (Bragge and Merisalo-Rantanen, 2009, Tuunanen, 2003), and where users play a large role in the provision of the value provided by the system (Iivari and Iivari, 2010). The involvement of the user can take on many forms, varying in involvement from seeking to understand an envisioned, fictitious user (Norman, 1986), to providing feedback for the system, (Bragge and Merisalo-Rantanen, 2009), and to actually providing direct input for the design (Iivari and Iivari, 2010). Direct user involvement, especially in a new product development environment can be
difficult (Iivari, 2006) because it is difficult to contact or identify potential users. Often representative users are selected to stand in for a larger group of users. Groups of users can participate simultaneously and efficiently in development workshops (Bragge and Merisalo-Rantanen, 2009) or focus groups (Bragge and Merisalo-Rantanen, 2009).

2. Requirements Specification Techniques

Many methods in requirements elicitation focus on the risks associated with highly complex requirements through the development of formal notations, or rules about writing and organizing requirements, and analysis techniques which allow for a precise reasoning about the requirements (Goguen, 1996). Formality can also aid in the identification of requirements by facilitating reasoning about requirements thoroughness (van Lamsweerde, 2000). Lamsweerde and other researchers have presented the notion of goals in formal analysis techniques as a means of assuring that requirements are complete. Notations can be more or less formal. Stakeholders from a business side of the project will be less involved in the development of requirements when formality is high, due to the high learning cost of communicating requirements formally (Dean et al., 1997). To leverage the extra work used to document requirements in the notation, they are often designed to be useful in later stages. Notations enable automated tools which can quickly perform analyses. Amongst the most prominent notations are the Unified Modeling Language (UML), which specify, amongst other items, use cases, objects, and data flow diagrams. This is amongst the most popular family of methods used to document, analyze and communicate requirements. UML use cases and scenarios allow a development team to consider users’ needs by including a representation which allows the team to reason about them.
3. Requirements Prioritization Techniques

Prioritization techniques support decisions about how to allocate resources to the development of software requirements. Mathiassen and colleagues (Mathiassen et al., 2007) note that some of these techniques are easy to use, such as the critical success factor (Byrd et al., 1992) method, and card sorting technique (Maiden and Rugg, 1996). Other techniques such as group supported EasyWinWin method (Boehm et al., 2001) or the Quality function deployment technique (Duggan, 2003) are more elaborate. We now discuss the last category of techniques, experimentation techniques.

4. Experimentation Techniques

Experimentation techniques advance understanding of requirements for software by developing prototypes or simpler versions of the software, as an “experimental” version of the software in order to surface understanding of the software. These techniques use software as a tool to communicate with users (Maiden and Rugg, 1996) and aid in reducing the risks associated with the identification of requirements that the user had not thought of or had not expressed adequately. The iterative nature of software developments methods such as the iterative development of software within Agile methods can be considered falling into this technique, as each iteration can be considered as a progressively advanced prototype.

2.5 The Value of User Group Requirements Elicitation Workshops in Requirements Development

We can now classify the group requirements brainstorming activity as primarily a knowledge elicitation activity, and primarily as a requirements discovery technique (See Figure 3). It is primarily a requirements discovery technique, given the heavy emphasis
of the activity on exploring the needs of users. The value, therefore, of group requirements elicitation workshops lays in the ability to discover many system related needs directly from a large number of users.

Such an activity would have been preceded by an effort to understand and scope the problem that the system would address, as well as an effort to negotiate and prioritize high level requirements amongst project stakeholders. The activity should provide enough knowledge about the needs of users that the development team can proceed to specify requirements. As was stressed by Chakraborty and colleagues (Chakraborty et al., 2010), previous requirements development activities may be visited as knowledge develops.

Figure 3: Focus of the Current Study: A Requirements Elicitation Technique with a Discovery and Specification Emphasis
The design of a requirements elicitation technique has implications for the overall software development process—in terms of the impact of the quality of the requirements as well as the potential impact of the technique on the way the software development process is conducted. A more complete set of requirements in the requirements elicitation stage will contribute ultimately to the quality of the requirements determination process. A more complete set of requirements will help developers to design the software correctly the first time, avoiding rework arising from small details of requirements left unexpressed. Testing efforts will also improve as the testing team will have a clearer picture of how the software is supposed to work as expressed in the requirements.

To improve the completeness of the elicited knowledge, the technique uses prompts designed to increase a user’s recall of knowledge. Understanding of how this technique influences the users’ behavior in this manner is presented in section 3. The technique will also be explained in further detail and evaluated in section 4.

3. Increasing Requirements thoroughness in User Centered Requirements Elicitation Workshops

3.1 Previous Research on User-Centered Requirements Elicitation Workshops

The literature contains several examples of stakeholders’ gatherings where they develop requirements. Joint Application Development has been utilized for many years as a packaged approach to developing requirements and software with all key stakeholders present (Liou and Chen, 1994). EasyWinWin is an approach for helping stakeholders to develop and negotiate a set of high level requirements amongst key stakeholders (Boehm
et al., 2001). As mentioned previously, Bragge and colleagues (Bragge and Merisalo-Rantanen, 2009) presented a method of eliciting requirements from users in GSS supported workshops.

However, studies comparing different group requirements development methods are difficult to find, especially empirical studies. There are a few notable exceptions. Dennis and colleagues (Dennis et al., 1996) found that when a requirements thoroughness problem was decomposed into separate parts and presented one part at a time, students generated a more complete set of requirements than those who were presented with the entire problem at the same time. In an unrelated experiment, Hickey and colleagues (Hickey et al., 1999) found that students generating requirements without specific guidelines for writing usage scenarios, or descriptions of a specific usage of the system, were more complete than those written by students who were instructed to give each scenario a number and title. The current study supplements these previous studies by focusing on a broader set of cognitive principles for designing requirements elicitation prompts, or directives given by analysts or facilitators to users, and by exploring the different effects of the prompts for collaborating and electronic brainstorming groups.

In order to be able to conduct this exploration, section 3.2 and 3.3 reviews previous research in requirements elicitation and group brainstorming respectively, to identify theoretical principles which can explain and predict performance of individuals and collaborating groups during a requirements elicitation brainstorming task. In section 3.4, the propositions to be tested in this study will be presented, along with corresponding hypotheses.
3.2 Previous Research in Requirements Elicitation Techniques

Requirements may be gathered from several sources such as documentation of existing processes, users, stakeholders, and existing software. However, much of requirements elicitation research focuses on the eliciting of requirements through communication with stakeholders (Dieste and Juristo, 2011). A key outcome of interest in this research is requirements completeness (Dieste and Juristo, 2011). Although much of the research presented in this section studies interviewing as a requirements elicitation technique, in many instances, the theoretical justification for the findings could apply to other requirements elicitation techniques, since they revolves around a foundational understanding of human cognition which is not limited to predicting the performance of interviewing techniques in eliciting knowledge. I present a summary of such findings in the following section in order to develop a theoretical foundation for the current study.

3.3 Overview of Human Cognition in Requirements Elicitation

Several characteristics of human cognition must be considered when attempting to improve the process of knowledge elicitation and more specifically requirements elicitation. The requirements elicitation process is a process with unique challenges. Knowledge elicitation encourages an expert, (i.e. a user who is expert at a particular task) to explain the steps in the processes taken as a job is performed. As an individual becomes increasingly expert at a task, procedures become increasingly automated and less reliant on conscious thought (Sweller et al., 1998). The knowledge of what is performed and why it is performed is forgotten. Once this knowledge is reconstructed, it
must also be presented to other stakeholders who have many different views of the software. Just as a mechanic, driver, and designer have different views of a car, a stakeholder and developers will have different views of the software (Orlikowski and Gash, 1994). This means that the developers may be concerned about details the stakeholder is unaware of (Orlikowski and Gash, 1994), or assumes are obvious (Maiden and Rugg, 1996).

The strategy of much of the requirements elicitation literature for overcoming the difficulties of the knowledge elicitation process has been to look to three areas of cognition: how knowledge is stored, how it is recalled, and how it is processed in relation to external stimuli. These topics will be discussed, and then the strategies which have incorporated them will be discussed in section 3.3.

**Knowledge Storage**

To improve the elicitation of knowledge, we must begin with an understanding of how knowledge is stored in memory. Is all knowledge stored in the same way? Are different strategies needed to elicit different types of knowledge? Are there aspects of knowledge storage that can be leveraged in order to improve the elicitation process?

Two major types of knowledge are prevalent in cognition literature as well as knowledge elicitation literature (Robillard, 1999): declarative and procedural knowledge. Most simply put, declarative knowledge is knowing what, and procedural knowledge is knowing how. Declarative knowledge involves the network of conceptual knowledge(Collins and Loftus, 1975). This can include knowledge of facts and the relationships between them. It describes the attributes or properties of objects. Procedural knowledge is knowledge which enables a person to act without conscious thought on a
recognizable situation to achieve an outcome. These conceptualizations of knowledge correspond to tacit and explicit knowledge, often presented in the knowledge management literature (Nonaka et al., 2000). The two types of knowledge are related, as one can see through the concept of expertise. Expertise is gained as the conscious steps taken to recognize and complete a task become increasingly automated and decreasingly part of a conscious activity where concepts are used to think through the problem (Sweller et al., 1998). Tacit knowledge might also be considered procedural knowledge as Nonaka has also conceptualized it as “knowing how” (Nonaka et al., 2000). Some knowledge, according to the ACRE framework (ACquisition of REquirements) (Maiden and Rugg, 1996), is taken for granted and it may be expressed, but the knowledge holder does not recognize the need to express it. This knowledge is often revealed when a stakeholder views a working software prototype and notices that a function used every day by the stakeholder, but unknown to an analyst or developer building the software, is missing.

Declarative knowledge may be semantic or it may be episodic (Robillard, 1999). Semantic knowledge is knowledge of the meaning of concepts, such as the meaning of “stakeholder” or “system.” While semantic knowledge is context independent, episodic knowledge is connected to context. An example of episodic knowledge could be an understanding of how often requirements change in a software development project. Declarative knowledge has been conceptualized to be stored in schema, or a loosely coupled network of conceptual nodes (Collins and Loftus, 1975). These nodes are connected by associations, often multiple associations, which allow a concept to be
connected to multiple other concepts at different times given the context of the use of the knowledge.

The network structure of declarative knowledge, while highly flexible, is fraught with imperfections and uncertainties. Even declarative semantic knowledge is not usually stored in a well-structured, complete state (LaFrance, 1992). When knowledge is initially stored, people tend to store only the gist of the knowledge. Many of the nodes in the schema are given “default” values. Knowledge may be based on assumptions that are not true (Brainerd and Reyna, 1992). Another difficulty with the flexible network of knowledge nodes is that multiple “versions” of the knowledge may be recalled, depending on the context. The knowledge, when it does surface, is shaped by the way the context is understood (Winograd and Flores, 1986). Further understanding of knowledge recall processes is needed to know how to develop prompts to elicit knowledge.

**Human Knowledge Recall**

Theories about knowledge recall focus on what surfaces from long-term memory when the mind is supplied with a stimulus. Recall is shaped by several tendencies in human thinking, including the tendency to be influenced by the stimulus when reconstructing knowledge. The stimulus could be a spoken question, an image, or a written word (Anderson and McCulloch, 1999, Ciranni and Shimamura, 1999, Cull et al., 1996).

Knowledge recall is not entirely under our control (Posner and Snyder, 1975). We cannot decide what knowledge is recalled. However, there are some general rules of recollection which are helpful when predicting knowledge recall. According to the ACT (Adaptive Character of Thought) theory of human cognition presented by Anderson and
colleagues (Anderson, 1996, Anderson et al., 2004), what we recall is determined to a large extent by relevance—relevance of the schema in the past as well as the relevance of the schema to the current stimulus. The goal resolving function of the mind is used to determine what knowledge is relevant and what knowledge is not (Anderson et al., 2004).

Beyond the general description of cognition presented by this model, there are some special situations which are relevant to requirements elicitation and knowledge elicitation in general. Certain types of knowledge are more likely to be recalled, regardless of the stimulus. Several studies in the requirements elicitation domain have illustrated recall differences between semantic or abstract knowledge and concrete or episodic knowledge. Episodic knowledge is much easier to recall (Borges et al., 1977), and is remembered much longer (Anderson, 1983). Larsen and Naumann (1992) found that analysts who first worked with a physical data flow diagram, and then worked towards an abstract diagram, generated a more complete set of questions for users than when the order of diagram completion was reversed. They theorized that those who begin with an abstract diagram had too few specifics to work with. Zmud and colleagues (Zmud et al., 1993) also found support for the ability of concrete thinking in facilitating the elicitation of knowledge using interviews.

From the previous discussion, we see that the nature of the stimulus, in conjunction with the relevance of the stimulus assessed by the goal resolving function of the mind (Anderson et al., 2004) determines the nature of the knowledge recalled. A notable example of stimulus on the recall of episodic knowledge is the effect of the wording of questions on recall illustrated by Loftus and Palmer (Loftus and Palmer, 1974). In their experiment, subjects viewed a traffic accident and were asked “how fast
the cars were going when they hit each other?” The word hit was replaced by “smashed” and other words. All subjects viewed the same material. However, when asked how fast the car was going, those who had words “smashed” or “collided” as part of the question reported higher speeds of the vehicles than those who had the words hitting or bumping used to describe the impact of the cars. This example was used by Marakas and Elam (Marakas and Elam, 1998) as evidence for the need to carefully choose the semantics and even the structure of interviewing questions. They used this careful structuring to test the effectiveness of a structured interviewing technique designed by Lauer and colleagues (Lauer et al., 1992) who found that using a pre-specified flow of information gathering resulted in generating accurate dataflow diagrams. Their results, along with the earlier findings of Agarwal and Tanniru (Agarwal and Tanniru, 1990) confirm the results of structuring in the requirements elicitation context.

**Human Information Processing (How Reasoning Takes Place)**

There are universally applicable constraints on human reasoning, as well as reasoning behaviors which pose challenges to the requirements elicitation task. Because requirements elicitation is such a memory intensive task, the requirements elicitation literature relies heavily on the cognitive science literature for understanding how items recalled from memory are reasoned about (Pitts and Browne, 2007). Browne and Ramesh (Browne and Ramesh, 2002) compiled relevant findings and theories from the cognitive psychology literature. Many of these are recounted here.

To begin with, all reasoning takes place in working memory (Baddeley, 1992a). While humans have nearly unlimited capacity to store knowledge, the amount of information they can be consciously aware of at the same time is limited. The number of
items which can be held in working memory at the same time is roughly a half dozen (Miller, 1956). However, these items can be very complex chunks of related concepts, or schema (Orlikowski and Gash, 1994), which have been stored in memory. This limitation in information processing has been considered by many to be the bottleneck of information processing.

Because of this limitation on human processing, humans have the tendency to satisfice or reason with a minimal amount of information (Simon, 1990). While humans are capable of using experience to reason in depth to make good decisions (Perkins et al., 1983), they will not unless they are motivated to do so (Kunda, 1990, Kuhn, 1991). Based on experience, humans tend to use a set of heuristics which help them make sub-optimal, but usually correct enough decisions. Even in the context of a requirements elicitation interview, when an analyst understands that a large degree of thoroughness and detail is desirable, the tendency (corrected by experience) is to prematurely cease the collection of further details in favor of a more simplified model (Pitts and Browne, 2004).

As an additional challenge, we give certain information cues additional salience or bias, without any tendency to analyze the amount of heed we give these cues. Examples of information cues taking on too much salience or biasing information decision include using recent events as a surrogate for all past events (Tversky and Kahneman, 1973), information that confirms our beliefs (confirmation bias), and using a representative or a small sample from a population to make inferences about the rest of the population (Tversky and Kahneman, 1974). These heuristics become problematic for accurate recall of knowledge because many of our mental models are incomplete.
(Orlikowski and Gash, 1994). We may use heuristics to fill in the gaps with incorrect information.

### 3.4 Increasing Requirements Thoroughness

Some of the findings related to the requirements elicitation interviewing context have already been stated above. However, a brief presentation of the findings is given here as an explanation of the direction in which these findings are pointing. The research in this area has a certain degree of coherence, since many cases studies build upon and confirm previous findings (e.g. the use of structuring in interviewing (Marakas and Elam, 1998, Larsen and Naumann, 1992, Agarwal and Tanniru, 1990)). Yet, it is likely premature to create a general theory of requirements elicitation, since this research is still exploring important characteristics of human cognition which apply to the requirements elicitation problem. However, solutions can be designed, and contextually based models can be built to predict the performance of elicitation techniques based upon the cumulative recommendations of this research. I summarize the recommendations below:

**Related to knowledge storage**

- Include techniques which encourage the elicitation of concrete, or episodic knowledge as this type of knowledge is more likely to be recalled (Moody et al., 1998, Zmud et al., 1993).

**Related to knowledge recall**

- Semantic and procedural structuring—the purposeful wording and sequencing of a technique is crucial since knowledge is reconstructed in response to the

- Use repetition and variation to increase recall (Moody et al., 1998, Pitts and Browne).
- Remove distractions to increase likelihood of recall (Moody et al., 1998).
- Begin with more general questions and work towards more specific questions to avoid selective recall (Appan and Browne, 2010).

Related to knowledge or information processing

- Prompt the analyst and stakeholder to reason clearly (check assumptions, reflect on answers given, summarize) and completely to overcome the effects of satisficing and poor reasoning tendencies (Pitts and Browne, 2004, Pitts and Browne, 2007, Browne and Rogich, 2001).

These principles will be referred to in section 3.4 when hypotheses are developed about the performance of individuals in brainstorming requirements when prompted using the principles.

3.5 Group Requirements Elicitation

Requirements Elicitation as a Brainstorming Task

In a workshop setting, the requirements elicitation task is similar to creative ideation or brainstorming tasks studied for the past several decades in the group brainstorming literature. In both cases, the key bottleneck to task success is knowledge recall and application to the problem setting. The focus is less on the arrangement and refinement of the recalled knowledge into a polished, refined product. There are crucial
differences between the creativity task and the requirements elicitation task which will be reviewed in the following section.

**Effects of Collaboration on Knowledge Activation**

One of the main purposes of bringing groups of user together is to allow them to build upon each other’s concepts through the exchange of ideas, as is the case in many collaborative requirements development workshops. An extensive body of research spanning more than 30 years has explored the costs and benefits of performing brainstorming exercises in groups as a means of creative problem solving. Much of this research came as a response to Osborn’s initial claim of increased ideation productivity when collaborating as groups as opposed to electronic brainstorming groups (Osborn, 1953). However, empirical findings show that individuals in electronic brainstorming groups may actually generate more unique ideas (i.e., the standard measure of productivity) than groups of individuals working together (Stroebe et al., 2010, Lamm and Trommsdorff, 1973). Early group research attributed the decrease in productivity to several causes including social inhibition (Camacho and Paulus, 1995), free riding (Diehl and Stroebe, 1987) and production blocking (Diehl and Stroebe, 1987). Several of these costs associated with group brainstorming can be eliminated with electronic brainstorming. Electronic brainstorming systems enable many individuals to submit ideas simultaneously to a shared list via a network of computers. Such a system allows individuals to participate anonymously, and therefore without fear of ridicule or retribution, increasing the willingness of individuals to submit new ideas (Connolly et al., 1990). Parallel input allows for ideas to be contributed simultaneously (Valacich et al., 1994).
In spite of the capabilities afforded by electronic brainstorming systems (EBS), the findings for productivity gains over electronic brainstorming groups are mixed. In fact, some studies report that electronic brainstorming groups outperform groups using EBS, even in conditions favorable to groups (Pinsonneault et al., 1999). To achieve the benefits of group work with an EBS, a large group (12 or more) would be beneficial (Dennis and Valacich, 1993). Smaller collaborating groups in this study did not achieve significant advantages. The lack of synergistic advantages in small EBS groups may be due to the fact that individuals in EBS collaborative groups are spending most of their time generating ideas and are not reading the ideas of other group members (Dugosh et al., 2000). Dugosh and colleagues showed that smaller groups (of four members) were able to outperform electronic brainstorming groups when they are explicitly asked to pay close attention to the ideas of others.

In order to pinpoint the likely causes of production blocking in the absence of social factors, Nijstad and Stroebe developed the two stage model of production blocking (Nijstad et al., 2003). Stroebe and colleagues (Stroebe et al., 2010) attribute most of the production loss in collaborating groups to cognitive production blocking. According to the SIAM (Search for Ideas in Associative Memory) model proposed by Nijstad and colleagues (Nijstad et al., 2003), idea production consists of two underlying processes: the search for a retrieval cue used to activate an image or schema, and the use of the image to generate ideas.

SIAM is based on the same envisioned model of the mind as divided into two major components, referred to in the requirements elicitation literature: long term memory, which has a limitless capacity for storage, and working memory which can
consider very few items at a time (Miller, 1956, Baddeley, 1992b). In order to be brought into working memory (Baddeley, 1992b), an “image” must be activated either from an external stimulus or internally through spreading activation (Collins and Loftus, 1975). When an image is activated in working memory, all of the closely linked associations become almost automatically accessible. The strength of these associations differs based on the relevance to the current retrieval cue (Anderson et al., 2004). The stronger the association, the more likely knowledge is to be recalled. Because the images or nodes of associative memory are often highly associated, another image will be activated, allowing the individual to follow a “train of thought” of semantically related images. If a train of thought is exhausted, a new search cue must be developed. Thus, the individual continues in the first process until they are unable to retrieve new ideas.

A train of thought may be disrupted if there is too much time between the activation of an image in working memory and production of the next idea. The activated image may become de-activated. External cues (e.g., the ideas of other collaborating individuals) can become the source of this delay as an individual is distracted from the production of ideas using their own current image. This will lead to a decrease in the length of a semantically related cluster of ideas. If the individual experiences many such delays, he or she will leave many clusters of ideas partially unexplored, leading to a decrease in production. SIAM was used to demonstrate how delays in an individual’s ability to continue a train of thought effected the length of a train of thought or cluster, and the unpredictability of the delays led to fewer clusters, and fewer overall ideas (Nijstad et al., 2003).
There is also substantial evidence for a disruption in the train of thought from the psychology literature investigating recall in groups. In an experiment performed by Basden and colleagues (Basden et al., 1997), collaborating groups recalled fewer items from learned lists than electronic brainstorming groups. This relationship held even when motivational factors such as social loafing were controlled for (Weldon et al., 2000). Basden and colleagues proposed that the reason for the inhibited recall in the collaborative recall session was that individuals were not able to follow their own retrieval strategies. The items recalled by other group members would render the individual’s retrieval strategies obsolete. Generally, individuals tend to recall items in clusters or within categories, instead of across categories. This may be partially due to the fact that within category recall requires less cognitive effort. However, this clustering behavior is not observed in groups. Another source of evidence for strategy disruption in collaborative recall arises from the observation that the effect of collaborative inhibition is minimized when individuals are instructed to restrict recall to one category (Basden et al., 1997). In this scenario, the individual is again free to pursue his or her own recall strategy. While collaborative brainstorming can disrupt idea production, it also enables an individual to produce ideas by providing a retrieval cue to activate a new image. SIAM predicts that stimulation from external cues or prompts (such as the ideas of others or prompts from the facilitator) can reduce the effort and time needed to develop a new search cue (Nijstad et al., 2003). In addition to the ideas of other group members, prompts from a facilitator may serve as retrieval cues to activate knowledge (Santanen et al., 2004). The effectiveness of these prompts depends on their ability to activate diverse knowledge. Thus, the exposure of individuals to the ideas of others has both a positive
and a negative impact on the individual’s ability to produce ideas. The loss occurs when idea generation from an image is disrupted by the individual reading the idea of another group member, leaving a “train of thought” unfinished. The gain occurs when the idea of another group member activates a new image from which to generate ideas.

4. Presentation of the Research Model

Just as creative brainstorming is a highly knowledge activation intensive activity, requirements elicitation will also depend on successful knowledge activation. Remember that a user has difficulty recalling needs, which often must be expressed at a level of detail that the he or she has not previously considered consciously (LaFrance, 1992). As a user becomes expert at a task, the procedures become increasingly automated and reliant on unconscious thought (Sweller et al., 1998). Users will also have the tendency to explain a system in very high level details. Requirements elicitation is a highly unstructured problem, and is therefore prone to satisficing behavior (Simon, 1971), or other heuristic strategies (Gigerenzer and Gaissmaier, 2011), where heuristics are used to assess the extent to which requirements have been completely and correctly gathered (Pitts and Browne, 2004).

Just as the order and semantics of questions asked can determine the accuracy (Marakas and Elam, 1998), level of detail (Agarwal and Tanniru, 1990), and the overall completeness of requirements generated in requirements elicitation interview settings (Pitts and Browne, 2007), prompts used in a requirements brainstorming session are likely to have the same effect on individuals. SIAM predicts that prompts increase
requirements thoroughness to the extent that they activate new images in memory, just as the requirements elicitation literature focuses on the activation of knowledge in memory.

**Prompting Guidelines**

For the purposes of this study, we focus on a few principles from the requirements elicitation research and used these principles to design a set of prompts that activate knowledge in a brainstorming setting:

**Scenario Building.** Scenario building elicits a more complete set of requirements by stimulating concrete or visual knowledge of the use of the system. When visualized in this manner, as opposed to an abstract manner, individuals are able to recall a more rich set of details (Zmud et al., 1993). Scenario building also helps the user think of the experience with a system from beginning to end in step-by-step details. Using scenarios, it is much easier to capture the experienced details as opposed to abstract facts.

**Repetition and Rephrasing.** When users are repeatedly prompted about the same information, the depth of details of their answers for a particular category increases (Nijstad et al., 2003). This has been shown to occur in individual and group brainstorming processes, as well as requirements elicitation processes for individuals (Pitts and Browne, 2007).

**Critical Reflection.** Finally, it is necessary to encourage individuals to reason about recalled knowledge, since humans have the tendency to leave assumptions unchecked (Kuhn, 1991). The suggestion to critically examine one’s previous work and assumptions is also a stimulus that activates knowledge that otherwise would not have been considered (Pitts and Browne, 2007). To generate a more complete set of requirements, it is important to think of the system in terms of the goals that the system is
trying to achieve (van Lamsweerde, 2001). If the pertinent goals of stakeholders cannot be achieved as specified, then requirements are missing.

These principles are reflected in the prompts displayed below (see Textbox 1). They represent a combination of scenario building, repetition and reflection.

**Timing.** To improve the completeness of requirements generated using EBS brainstorming, the prompts from Figure 4 should be administered by a facilitator during the prompting session in large enough intervals (Santanen et al., 2004) that allow for individuals to have sufficient time to continue recalling knowledge in a train of thought (Bragge and Merisalo-Rantanen, 2009). The prompts in Figure 4 are used in the experiment.

```
1. Look at the requirements written. What other requirements or features do they make you think of?

2. As you look at the features described on the page, think what they enable the user to do. What will the user do before or after? Are there features to support those activities? Try to think from the beginning to the end of your experience with the website.

3. Look at the features in the list. Are there features missing that would need to be included to support those features?

4. Think about the goals that the features of the stories support. For example, a user may want to manage his profile. What functionality is needed to support this goal? What other features would be needed to support those goals?

5. Are there any details missing from the user stories on your list? Elaborate on information that is missing from the features.
```

**Figure 4: Requirements Elicitation Prompts**
Characteristics of the Requirements Elicitation Task

The prompts rely heavily on a critical difference between the requirements elicitation task and the creative idea generation task for their effectiveness—they more heavily leverage an individual’s existing knowledge. The strength of exploratory requirements elicitation techniques arises from their ability to leverage the knowledge that users already possess. These techniques are useful for the development of systems which are not overly complex, thus requiring extensive modeling, or not so different from systems or tasks which the users are familiar with that their expertise becomes unhelpful.

A major difference between the creative idea generation task and the requirements elicitation task is that the prompts rely on the relationship that exists between the knowledge items generated. While ideas generated in a creative brainstorming session tend to be related by association, reflecting spreading activation (Collins and Loftus, 1975), requirements can also be related to one another in other ways. For example, requirements can be related in that they must work together to meet a goal (van Lamsweerde, 2001), or complete a certain task using a system (Jacobson, 2004). Requirements also correspond in several ways to existing objects in the problem domain that they address, such as a process or a set of information needed to perform a particular task. Commonly used modeling techniques such as use cases and goal oriented requirements notation reflect this relationship between requirements.

Similar to the case of creative idea generation, an individual who is generating requirements will tend to use an image to activate closely related knowledge in a train of
thought, since an individual’s knowledge relevant to requirements is not necessarily stored any differently than knowledge used for a brainstorming task (LaFrance, 1992, Robillard, 1999). However, in addition, individuals may also use relationships between requirements as well as knowledge about the facts related to the requirements as search cues to activate new knowledge. For example, requirements may describe a system which will be used to buy and sell merchandise in an online marketplace. One of the prompts in Textbox 1 encourages an individual to reflect upon one of the requirements already written and continue to describe a scenario using the system the requirements describe. Reasoning about which step should occur next acts as a retrieval cue, allowing the individual to activate new knowledge to describe the next step in the scenario, and any implications for the remaining requirements.

Because individuals in electronic brainstorming groups are (a) only exposed to the prompts mentioned above and not to the requirements generated by other users, and (b) prompts are adequately spaced so that individuals are unlikely to prematurely abandon an activated image, we hypothesize that:

**H1: Prompted electronic brainstorming groups will generate a more thorough set of requirements than unprompted electronic brainstorming groups.**

Prompted individuals in electronic brainstorming groups will also perform better than shared requirements brainstorming groups for several reasons. First, unless prompted to do so, shared requirements brainstorming groups will tend to be satisfied with fewer requirements describing the system, just as electronic brainstorming groups would be. Second, the requirements generated by other group members are more likely to inhibit recall of knowledge instead of activate new knowledge. Since requirements
describe a constrained solution set, a requirements elicitation task more resembles a free recall task than a creative brainstorming task. It is therefore more likely that exposure to the ideas generated by others will result in the collaborative inhibition experienced in free recall tasks (Basden et al., 1997). Requirements generated by other group members will also inhibit recall because individuals may describe requirements for a system which are not compatible with their own. We thus hypothesize that:

**H2: Prompted electronic brainstorming groups will generate a more thorough set of requirements than unprompted groups which share generated requirements.**

The effect of the interference from the requirements generated by others will become apparent when the completeness of requirements generated by prompted individuals in electronic brainstorming groups is compared to that of individuals in collaborative prompted groups. The limited effectiveness of requirements generated by other group members as stimuli for the activation of knowledge will be due to increased recall inhibition. Additionally, the effectiveness of the prompts given by the facilitator to the group will also be limited in effectiveness because they rely even more heavily on group members having a common conception of the website. For example, one individual may have a different idea of what goals the requirements of the website should support. Assuming no effort to come to a shared understanding of the goals of the website previously, the prompt in Figure 4 pertaining to goals would therefore be less effective in activating the individual’s knowledge. The individual will prematurely abandon exploration of the activated image because they are reluctant to submit a requirement which is not compatible with another group member’s goals. We therefore hypothesize that:
H3: Prompted electronic brainstorming groups will generate a more thorough set of requirements than prompted shared requirements electronic brainstorming groups.

Groups which have access to requirements but which are not prompted to consider the user stories generated by other users will not have the same disadvantage as interacting groups which are prompted to do so. Groups sharing requirements which consider the user stories generated by others at their leisure are able to integrate and build off of the requirements at their leisure. They will not suffer from recall inhibition because they will tend to draw upon the requirements of others only when necessary, thus stimulating activation of new images. These groups will also experience less dissonance when thinking about the compatibility of the user stories of others since they are never prompted to think holistically about the site. We therefore hypothesize that:

H4: Unprompted shared requirements groups will generate a more thorough set of requirements than collaborating prompted groups.

5. Method

5.1 Participants

207 students from two Midwestern universities participated in the experiment, forming 59 groups of three and four individuals. Students pursued a variety of majors, predominantly business and psychology degrees, and the majority were undergraduates (the sample contained 10 graduate students). 64.1% of the students were female and 35.9% were male. The median age of the students participating was 19. A few were over the age of 30, with the maximum age at 55.
Each group was randomly assigned to one of the four treatments. Each treatment contained an approximately equal number of three and four member groups. Some other research suggests that group size, does not make a difference, unless group members are reading each other’s ideas (Ziegler et al., 2000). When groups are reading each other’s ideas performance of collaborating groups has been found to outperform electronic brainstorming groups in settings where both had a group size of four (Dugosh et al., 2000). A post-hoc T-Test comparing the requirements generated by groups of size three and groups of size four was conducted. The results of the T-Test were not significant ($P=0.06, T=1.85, d.f. 54$). Participants were placed in treatments by choosing a computer with a pre-assigned session of a web-enabled Group Support System (GSS).

### 5.2 Procedure and Tasks

The experimental task consisted of brainstorming user stories for an online textbook exchange system. This task was deemed sufficiently complex such that users would be able to generate requirements for a full 45 minutes and also because an online textbook exchange is a subject that is highly familiar to college students. User stories were generated and captured electronically using GroupSystem’s ThinkTank® version 2.4 group decision support software in all four treatments. Experiment personnel obtained informed consent and demographic information using a pre-survey.

The experiment was a between-group design with two treatments in two conditions (2X2): unprompted / prompted and collaborative / electronic brainstorming groups (see Figure 5). Each team was randomly assigned to one of the conditions and the session began. The facilitator first handed an informed consent statement to each participant (See Appendix D). The facilitator then introduced the task and the book
exchange website and then directed the participants to read a written description of the 
book exchange scenario. This description included (a) a description of the task, (b) an 
explanation of the key components of a valid user story, and (c) multiple examples of 
good user stories from the perspective of a student buying and selling on the website, a 
professor, and a website administrator. The task description given to each participant is 
presented in Appendix A. Once all of the team members were ready to continue, 
additional instructions and clarifications were given as necessary. Participants were given 
a final chance to ask questions of experiment personnel before the requirements 
elicitation task began.

The instructions began with a “generic prompt” from the experimenter indicating 
that subjects were to generate as many user stories as they could over a forty-five minute 
time period. In the unprompted groups, both electronic brainstorming and collaborative, 
no additional interventions were given and groups were left to create user stories on their 
own. In the prompted conditions, additional prompting and instructions were given. The 
facilitator had five scripted prompts (those presented in Textbox 1). The first prompt was 
given after 10 minutes and the remaining prompts were delivered at approximately seven-
minute intervals, resulting in a 45-minute requirements elicitation session. No scripts or 
prompts were given via the brainstorming software. The participants were aware of a set 
period to brainstorm. However, the vast majority of participants had finished 
brainstorming well before there end time.
Unprompted Electronic Brainstorming

Unprompted Shared
User Stories Electronic Brainstorming

Prompted Electronic Brainstorming

Prompted Shared
User Stories Electronic Brainstorming

Figure 5: 2X2 Experiment Design

Unprompted Electronic brainstorming Groups: Unprompted electronic brainstorming groups did not have access to the user stories generated by other members of their group. They were not even aware that their user stories would contribute to the set of user stories generated by other members of their group. They were not made aware that they were working as a team in any manner.

Unprompted Collaborative groups: Unprompted collaborative groups had access to the user stories generated by other members of their group but generated ideas on separate pages. The different pages were accessible by clicking on “buckets” which represented the electronic page. The buckets could be accessed by the other users. They were not encouraged to read the ideas of others by the facilitator.

Electronic brainstorming Promted Groups: In the electronic brainstorming-prompted groups, participants were told to look at the requirements that they had already
written themselves and apply the prompt given. The individual used the prompts to examine his or her own user stories, and then contribute several user stories based on that image activation. These subjects were also not made aware that they were working as a team in any manner.

**Collaborative Prompted Groups:** In the collaborative-prompted group conditions, each individual was assigned a bucket (buckets represented separate electronic pages accessible by the entire group) to apply the prompt given. With each new prompt, individuals were moved to a new bucket so that each individual would have to read all user stories in that bucket up to the time the prompt was given. Individuals in collaborative groups were free to explore other buckets after beginning with their own bucket. In the collaborative prompted task, the individual is using the prompt to examine possibly his or her own user stories as well as the user stories of other group members. He or she could generate several user stories before going to the next bucket. The next bucket contained the requirements generated from the previous prompt by the individual working in the bucket.

The prompted electronic brainstorming and prompted collaborative tasks are equivalent in that in each case, the individual is reading roughly the same quantity of user stories. The tasks are different in that the individual in the electronic brainstorming group prompted treatment is reading his or her own requirements, whereas the individual in the collaborative group prompted treatment is reading the requirements of others as well as his or her own.
5.3 Measures

To measure thoroughness of requirements generated, a set of 108 requirements was created from an initial set of requirements obtained from a real book exchange website which were validated by a software development expert, and merged with a set of compatible requirements obtained through categorizing roughly half of the user stories (3942 user stories) generated by participants. This resulted in a set of 108 requirements in 19 categories. If a group generated a higher total number of requirements within this list, then the set of requirements were considered more complete. The set of requirements included an “out of scope” category which was used to remove user stories which were not compatible with the scope description given to the students (e.g. “As a seller and buyer I would like the website to have a payment system so that I can easily pay and be paid” should not be included since the instructions state that payments should not be made), or which did not contain any actionable details (e.g. “As a website administrator, I want my website to get more views with each passing day so that I get more revenue”). These exclusions helped to ensure that the set of user stories generated by groups reflected potentially valid requirements. The categories are listed and the requirements within each category are briefly described in Appendix B. Two coders blind to treatment assignment coded the user stories generated by each group, using the coding rules in Appendix C, to obtain a total number of requirements for each group. Each coder used these rules independently to place a requirement as covering one of the 108 requirements in the 19 categories. Following individual coding, the two coders would come together and compare which categories were associated with the user stories generated by the group they had coded separately. When one coder found user stories associated with a
requirements category and the other did not, they would discuss whether or not the user stories found were candidates, until consensus was reached between coders. An average agreement level before reaching consensus of 88.7% was reached. A Cohen’s Kappa coefficient was calculated for agreement to assess the level of agreement between coders which was due to chance. The resulting kappa coefficient was .712 which is considered ‘substantial’ agreement (Landis and Koch, 1977).

*Raw User Story Contributions* are calculated as the total number of items input by all members of a treatment group into the brainstorming tool during the brainstorming session.

### 6. Data Analysis and Results

The analysis of the data collected consists of several tests which illustrate the effects of collaboration and prompting. The most critical concerns with the testing of the three hypotheses which test the effect of collaboration and prompting on requirements generated. The comparison of the means of requirements generated by participants in each of the treatment groups are discussed in section 5.1. In section 5.2, a similar test compares the means of user stories generated by teams in each treatment group. Additional analyses not related to the hypotheses of this study are included in Appendix E. The analyses presented in Appendix E explore the individual effectiveness of each prompt used in the prompting conditions.

#### 6.1 Hypotheses Tests
Before tests were run to compare treatment means, the requirements generated by each group were analyzed for outliers, equal variance, and normality.

**Descriptive Statistics:** The descriptive statistics for requirements generated are reported below in Table 1:

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Mean:</th>
<th>Standard Deviation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean:</td>
<td>40.64</td>
<td>11.28</td>
</tr>
<tr>
<td>Minimum:</td>
<td>13.00</td>
<td></td>
</tr>
<tr>
<td>Maximum:</td>
<td>66.00</td>
<td></td>
</tr>
</tbody>
</table>

*Table 1: Descriptive Statistics for Features Generated for All Treatment Groups*

Descriptive statistics for the requirements covered by treatment group is presented in Table 2. Figure 6 shows the descriptive statistics as a bar graph.
Table 2: Mean and Standard deviations of Requirements Generated by Each Treatment Group

<table>
<thead>
<tr>
<th>Treatment Type</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic brainstorming Prompted</td>
<td>47.15</td>
<td>7.95</td>
<td>13</td>
</tr>
<tr>
<td>Collaborative Prompted</td>
<td>36.4</td>
<td>12.25</td>
<td>16</td>
</tr>
</tbody>
</table>

Figure 6: Descriptive Statistics

Normality

The normality of requirement data in each treatment group was assessed visually using the following histograms. No set of data had any significant departures from normality.
Variance

A significant ANOVA $F(3,54)=2.82$, $p=0.047$, indicated that the null hypothesis of the homogeneity of the treatment groups could be rejected. Then, independent samples T-Tests were used to compare the prompted electronic brainstorming treatment group mean to every other treatment group mean. As part of each T-Test, a Levene’s Test for equal variance was conducted. None of the tests returned a significant result.
Hypotheses Tests Results

The results of the T-Tests for each hypothesis are reported in Table 3. The results support two of the four hypotheses (H2, H3) at the \( P < 0.01 \) level for one-tailed t-tests. H1 and H4 were not supported. Prompted electronic brainstorming groups were able to outperform prompted groups (H2) and unprompted collaborative groups (H3), but not unprompted electronic brainstorming groups (H1). Unprompted collaborative groups did not outperform prompted collaborative groups (H4).

<table>
<thead>
<tr>
<th>Hypothesis Description</th>
<th>D.F.</th>
<th>T-Statistic</th>
<th>P-Value*</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1: Prompted Electronic brainstorming vs. Unprompted Electronic brainstorming</td>
<td>24</td>
<td>1.20</td>
<td>0.240</td>
</tr>
<tr>
<td>H2: Prompted Electronic brainstorming vs. Collaborative Unprompted</td>
<td>27</td>
<td>2.65</td>
<td>0.0131</td>
</tr>
<tr>
<td>H3: Prompted Electronic brainstorming vs. Prompted Collaborative</td>
<td>27</td>
<td>3.32</td>
<td>0.0030</td>
</tr>
<tr>
<td>H4: Unprompted Collaborative vs. Prompted Collaborative</td>
<td>29</td>
<td>.2715</td>
<td>0.78</td>
</tr>
</tbody>
</table>

Table 3. Results of t-test by hypothesis for requirements covered

*All significance values are for one-tailed tests
6.2 Comparison of Raw User Stories Generated

All though the hypotheses tests compared the requirements generated by each group, an addition analysis of the raw user story contribution data was conducted. The descriptive statistics for contributions hit is reported in Table 3. An ANOVA test for multiple means in the user story contribution data was not significant. However, the results of pairwise t-tests are similar to the comparison of requirements generated for the comparisons between the electronic brainstorming prompted treatment group and other groups—t-tests are significant for one-tailed comparisons at the $P < 0.05$ level for electronic brainstorming prompted vs. electronic brainstorming unprompted means and electronic brainstorming prompted vs. collaborative prompted means and a difference in means which was significant at the same probability level for electronic brainstorming prompted vs. collaborative prompted means. A prompted electronic brainstorming vs. prompted collaborative mean comparison was not significant.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic brainstorming</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unprompted</td>
<td>64.66</td>
<td>28.81</td>
</tr>
<tr>
<td>Collaborative Unprompted</td>
<td>72.43</td>
<td>31.89</td>
</tr>
<tr>
<td>Electronic brainstorming</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prompted</td>
<td>86.69</td>
<td>24.07</td>
</tr>
<tr>
<td></td>
<td>67.37</td>
<td>44.12</td>
</tr>
</tbody>
</table>
Collaborative Prompted

Table 4. Descriptive statistics for number of user stories generated by treatment

<table>
<thead>
<tr>
<th></th>
<th>D.F.</th>
<th>T-Statistic</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prompted Electronic brainstorming vs. Unprompted Electronic brainstorming</td>
<td>23</td>
<td>2.08</td>
<td>0.03</td>
</tr>
<tr>
<td>Prompted Electronic brainstorming vs. Collaborative Unprompted</td>
<td>27</td>
<td>1.33</td>
<td>0.1941</td>
</tr>
<tr>
<td>Prompted Electronic brainstorming vs. Prompted Collaborative</td>
<td>27</td>
<td>1.41</td>
<td>0.1688</td>
</tr>
<tr>
<td>Unprompted Collaborative vs. Prompted collaborative</td>
<td>29</td>
<td>.3677</td>
<td>.7157</td>
</tr>
</tbody>
</table>

Table 5. Results of t-tests for user stories generated for selected comparisons
* All significance tests are one-tailed

7. Discussion

The results of the study provide strong evidence that requirements-based prompts result in more requirements generated when used when individuals are not interacting to develop a more thorough set of user stories. The comparison of the number of unique user stories generated by the electronic brainstorming prompted individuals in to the unique user stories generated by the prompted interacting groups (H3) demonstrates this
difference in stimulus benefit directly. When individuals in interacting groups are not prompted to examine and incorporate the requirements of others, the requirements of others still do not provide a prompting benefit (H2). The benefit of the prompts to individuals was not demonstrated through the comparison of a prompted electronic brainstorming group to an unprompted brainstorming group. However, prompted individuals were able to generate a higher quantity (including duplicates and out of scope users) of contributions than unprompted individuals. It is likely that, with a larger sample size, a significant difference between prompted and unprompted electronic brainstorming groups could be found. It is also possible that with further improvements of the prompts, that a positive difference could be observed.

The superior performance of prompted individuals over prompted collaborative groups in the current study demonstrates a diminishing effectiveness of the prompts when used in a collaborative setting. The results of the experiment should be considered in light of related research (Santanen et al., 2004, Dugosh et al., 2000) which demonstrate a positive instead of a null effect, as was demonstrated by the inability of our study to reject (H4). The differences highlight the need to examine the differences between the studies. The important differences will be highlighted here.

On the surface, the lack of benefit from prompting in interacting groups in this study contradicts the findings of Santanen and colleagues where prompted collaborating groups generated more unique ideas than prompted electronic brainstorming groups. However, a closer comparison of the tasks and prompting treatments presented to the participants highlights the differences between creative brainstorming tasks and requirements elicitation tasks, especially since the tasks in both studies are otherwise very
similar. Group sizes of four were used in the study of Santanen (three or four in this study), the brainstorming time was roughly equivalent (40-45 minutes), and the tasks were both appropriate for novice, student subjects. In Santanen and colleagues’ study, prompts were administered once every two minutes over the entire duration of the 40-minute brainstorming session, whereas in the current study, prompts were only administered once every seven minutes over a 45-minute brainstorming session.

It is possible that the difference in the semantics of the prompts used in the studies is responsible for the difference between the findings of the current study and the study of Santanen and colleagues. All the prompts in the current study encouraged individuals to reflect on previous prompts whereas only one out of four prompts in Santanen and colleagues’ study encouraged reading. It appears that there is support for the idea that prompts which encourage reflection about existing ideas, when used to process the contributions of others, has a significantly different effect than prompts which encourage an individual to simply generate ideas in relation to a particular topic or goal such as those used in the study of Santanen and Colleagues which encourage participants to suggest solutions that are inexpensive, easy to implement, can be implemented quickly, and that please everyone. It is likely that these prompts enable individuals brainstorming both with and without interaction.

Reading the ideas of others, however, has been shown to be beneficial to collaborative electronic brainstorming in the study of Dugosh and Colleagues (Dugosh et al., 2000). Dugosh and Colleagues found that interacting brainstorming groups of four individual using GSS stations generated significantly more unique ideas when they were encouraged to pay attention to the ideas generated by others when compared to
interacting groups which were not encouraged to pay attention to the ideas of others through the use of a post-test where the ideas of others would be. These interacting groups also did not perform significantly better than brainstorming groups working separately.

The findings of this study point out a third area of investigation in the use of prompts to improve collaborating groups. Not only should the use of prompts and the extent to which they encourage an individual to read the prompts of others be explored, but these studies should also consider the nature of the requirements development vs. the brainstorming task. The SIAM model (Nijstad and Stroebe, 2006) and the research in collaborative recall inhibition (Basden et al., 1997, Weldon et al., 2000) provide an initial lens for exploring the task nature of the task. First, SIAM predicts a loss in productivity when stimuli, such as the ideas generated by other individuals are ineffective or cause an individual to prematurely abandon exploration of an image or set of related concepts. A set of requirements is likely to be much more bounded than a set of brainstormed ideas. A set of requirements is also likely to be more related—all requirements describe the same concept or potentially a set of related concept. If the requirements are familiar with the object that the requirements describe, as is likely the case in the current experiment, then the images associated with the website are likely to be highly interconnected. This may make the contributions of others incompatible with the individual’s notion of the website, thus making the contributions of others ineffective at activating an individual’s knowledge of the website. Alternatively, or additionally, the contributions of others may encourage individuals to abandon a train of thought related to the website prematurely because they feel they are no longer compatible with the website, for example.
An example may clarify how familiarity with a website might create a situation where incompatible user stories generated by another user might either be ignored or interfere with an individual’s conceptualization of a website. An individual may have had experience buying books on Half.com where transactions take place online and buyers are not restricted. Another individual may have had an experience with a website more like Craigslist where no transactions take place online and buyers are preferably local. If the individual thinking of the Half.com website encounters contributions such as assuring that the buyer is a student or that the meetings take place in a safe place, then the first individual might be confused as to the overall vision for the website, since the prompts in this experiment encourage the individual to think of requirements related to the one they are looking at. At least such contributions will be ignored by the individual and will not activate further knowledge. Because familiarity with the website was not measured, and no effort was made to measure a train of thought, these suppositions require more research to substantiate. This will be discussed in further detail in the next section.

As a final note of discussion, this study underlines the importance of cognitive principles to requirements elicitation research. Applying understanding of how knowledge is stored, recalled, and processed in working memory will result in measurable requirements elicitation outcomes. This has been demonstrated previously in requirements elicitation literature (Pitts and Browne, 2007, Zmud et al., 1993, Marakas and Elam, 1998, Moody et al., 1998). We used multiple principles to design each of our prompts including repetition and rephrasing, critical reflection, and scenario building which drew upon cognitive principles highlighted in the literature just mentioned.
Drawing upon the analysis in Appendix E, it appears that the prompt which encouraged scenario building was the most successful for individuals. Our study highlights that this principle, along with the other principles mentioned may not always be applicable in collaborative settings. Future research will seek to describe the circumstances where cognitively designed prompts will improve requirement generation performance in collaborative settings.

### 7.1 Limitations and Future Research

The current study highlights several areas of future research. Some of these areas stem from the limitations of the experimental design of the study, while others stem from limitations of its applicability to the requirements development domain. Finally, a third area of future research focuses on investigating crucial independent variables which influence activation of knowledge and trains of thought in SIAM.

**Study Design Limitations**

We did not test the prompts for effectiveness. One of the prompts—the prompt at 17 minutes which encouraged individuals to think of the steps before or after a requirement was much more helpful than the other prompts in the study, as is illustrated in Figure 7 in Appendix C. While we suppose that prompts which encourage reflection and reading of previously written prompts will continue to prove ineffective, it would be useful to be able to show that the prompts were indeed effective.

On a related note, the sample size in our study was not likely adequate to detect a significant difference between prompted and unprompted individuals. This would have provided some demonstration that the prompts were effective.
It is likely that social comparison effects played some role in the amount of contributions generated by each individual since individuals generated requirements in the same room and could hear whether or not other individuals were typing. This social comparison effect likely acted on all treatment groups. This effect likely dampened individual differences in prompting performance—some individuals who would have finished earlier persevered because they felt behind others who were still typing. Other individuals might have stopped generating ideas prematurely when they realized that everyone else had stopped typing. Future research could determine if there was a significant in-the-same-room effect.

Along with social comparison effects, social loafing must be considered as well. The possibility that individuals in collaborative groups were engaging in social loafing must be considered, and as must the possibility that this counted at least partially for the difference between prompted electronic brainstorming and prompted collaborative groups.

Individuals in all treatments may not have understood clearly that they were supposed to generate a complete set of requirements. More unique requirements might have been generated by collaborating groups and individuals if participants in each treatment group were given, for example, a rough outline of what a complete set of requirements should look like.

Another potential confound of the study may have arisen from the additional cognitive load that individuals who were required to read and interpret the user stories of other contributors. Individuals using the prompts on their own requirements may not even have to read their own requirements to come up with new ones. If the cognitive
load is not directly related to image activation, this may be an unfair advantage. There is some evidence from the study that this is not a factor. For example, the most effective prompt encourages individuals to think of requirements which involve thinking of steps taken before and after the steps implied by the requirement in the requirement. This would involve some reading and consideration on the part of the prompted individual as well, yet it results in the most user stories generated. The other argument against this is the amount of time needed to read the requirements vs. the time needed to generate requirements. A brief examination of the time stamp data mentioned in Appendix C also shows that the time to generate ideas was likely to be ample since there were large gaps of time after most of the prompts had been written. Future research could also more precisely measure the cognitive effort spent analyzing the contributions of others using eye-tracking technology.

Continuing this thought, perhaps one of the key limitations of the study is that no effort was taken to measure trains of thought or activation. Unlike the studies mentioned in Stroebe and Colleagues (Stroebe et al., 2010) we did not measure clustering of ideas or attempt to measure trains of thought in any fashion. We are therefore left to guess the extent to which extent the production loss is due to unsuccessful activation or from premature abandonment of a train of thought. Careful exploration of the data and collection of new data, perhaps with the addition of post-task interviews where individuals are asked about the extent to which the user stories submitted by others were detrimental to their own understanding.

This study design could incorporate procedures from group brainstorming research which have been found to allow collaborating groups to outperform groups
brainstorming as individuals, such as the study by Paulus and Yang (Paulus and Yang, 2000) which found that when groups collaborated before an individual brainstorming session, those groups brainstormed more unique ideas than groups of individuals which had never met together. This procedure should be incorporated into future research in comparative requirements elicitation.

**Enriching SIAM**

SIAM presents a process of image activation and idea generation to explain idea generation in groups, but, as it is described in Stroebe and colleagues (Stroebe et al., 2010), it does not elaborate on how images are successfully activated, nor does it elaborate on the mechanics which sustain an individual in the exploration of a train of thought. It does explain that the cognitive load of focusing attention on the idea of another will compete with the attention needed to keep an image activated, with too much cognitive load resulting in the image becoming lost to attention.

The current study presents a case where it becomes necessary to describe the nature of the stimulus used to activate an image. Future research should determine whether or not the diminishing effectiveness of the prompts is due to recall inhibition, for example. If recall inhibition cannot account for this diminishing effectiveness, then there are several possible causes for the poor performance of the prompts as a stimulus in groups. The diminishing effectiveness of the prompts may originate from an unwillingness or an inability of an individual to build on a statement made by another individual which is ambiguous or incompatible.

Additionally, the nature of the images activated in a requirements elicitation task and the mechanics of maintaining those images in active memory likely differs greatly
from the images maintained during a creative brainstorming task. For example Nijstad and colleagues (Nijstad et al., 2003) measured the sustained use of an activated image by determining the extent to which the brainstorming individual generated ideas within the same semantic category. It is possible that the images activated when individuals generate requirements are not semantic, but are perhaps episodic. Perhaps an individual relives an experience with a website and that particular experience can be used to trigger related experiences. The nature of these images, and the extent to which they are more or less vulnerable to the interruption of the requirements generated by others is a question brought to the forefront by the current study.

The exploration of the effect of group collaboration and prompting should be expanded into a program of research. The future research previously mentioned suggests a couple studies. The first would replicate the conditions mentioned in this dissertation, but control for social loafing, perhaps by providing a strong motivation for both collaborating and non-collaborating prompted groups to perform at their best (e.g. a monetary prize for the most requirements). Another study would control for differences in understanding of the system between individuals by giving each participant a detailed description of the system before beginning the brainstorming, and then assessing their understanding of the system with a questionnaire. If the scores on the tests are equivalent between treatment groups, then common understanding of the system could be assumed.

Beyond these two studies, exploratory research with smaller sample sizes should be conducted to identify the optimal conditions for generating a complete set of requirements in a collaborating group. There are many possible modifications to the task which have been shown to improve the performance of collaborating groups in group
creativity research. For example, Paulus and Yang (Paulus and Yang, 2000) found that collaborating groups out-performed electronic brainstorming groups at generating a unique set of ideas when they were allowed to generate ideas separately following an initial session where individuals were exposed to the ideas of others. Reiter-Palmon and colleagues (Reiter-Palmon et al., 2008) presented several propositions about the creative solution generation performance of collaborative teams within a previously developed cognitive framework of creativity (Mumford et al., 1994) which could provide an outline for this research.

Some strategies for improving the requirements brainstorming problem originate from knowledge unique to the requirements development domain. A large body of research has studied the way that the rules for structuring the expression of requirements impacts an individual’s ability to express (Wand and Weber, 2002), recall (Marakas and Elam, 1998), and comprehend (Khatri et al., 2006) requirements. This research community would likely be interested in studies which demonstrated the effectiveness of requirements structuring for requirements generation tasks.

**Practical Applicability Limitations**

Our study only improves requirement generation thoroughness and does not represent a complete product of the requirements elicitation and verification process. Requirements elicitation methods such as interviews and workshops as well seek not only for users and other stakeholders to communicate knowledge to analysts, the analysts must also validate their understanding of the stakeholders’ requirements, (Browne and Rogich, 2001). Further, these requirements must be agreed upon by all stakeholders and prioritized according to their importance (Boehm et al., 2001). The results of this study
must be seen as only improving a small step in the overall requirements development process.

Our findings also apply to a workshop process which are likely to be used in larger projects and projects which use the “waterfall” method of developing software. Because of the increasing adoption of Agile software development methods, the applicability of workshop methods to Agile software development must be considered. The workshop method may be overkill for smaller projects which do not benefit from the involvement of many users or do not involve the production of many user stories up front. However, a workshop may be an effective way for a team to develop an up-front backlog of user stories. Consideration of the thoroughness of user stories generated would be helpful in this situation.

Future research should also see if the results apply within the constraints of specific workshop methods such as Commonality and Variability analysis, Joint Application Development and EasyWinWin. The specific semantics of each of these methods may help collaborating groups

The findings of this study may not generalize to all sets of requirements. First, the envisioned system in another context may be less understood and less bounded than the book exchange system presented to users in this study, reducing the likelihood that group members will generate requirements in a similar manner. The results would also likely be different if individuals had a similar understanding of the system, which would be the case if users brainstorming were designing a replacement for the system which they had previously used.
We did not measure the extent to which duplicates were reduced by any of the treatments. It is possible that collaborating groups could be considered more efficient than non-collaborating groups if the number of duplicates in non-collaborating groups were significantly more than the duplicates found in collaborating groups.

It would also be interesting to compare amongst the different treatment groups the extent to which each did or did not go out of scope. It may be possible that collaborating groups produce a lower rate of false positives since they are more aware of each other’s prompts.

### 7.2 Practical Implications

Our study has directly implementable suggestions for practitioners. First, we have demonstrated the effectiveness of a set of requirements elicitation prompts which can be used in the facilitation of GSS supported requirements elicitation workshops. Because these prompts are domain independent, they may be used to generate requirements for a variety of systems. We may also advise facilitators that the prompts may be less effective if individuals are exposed to the requirements generated by others. These prompts could also be used in other requirements elicitation settings such as requirements negotiation (Boehm et al., 2001) or early Joint Application Development sessions (Boehm et al., 2001)

### 7.3 Conclusions

We have demonstrated that prompts based on cognitive principles used in user-centered requirements elicitation workshops can improve the thoroughness of generated by users. If previously untrained users are guided in their efforts to contribute
requirements, they may be able to contribute requirements of a higher quality. We provide support for this conclusion by demonstrating the increased thoroughness of requirements generated by users who were exposed to prompts which encouraged them to reflect on previously generated requirements and consider which requirements could be missing from a more holistic view of the system. We find that the effectiveness of the prompts used in this study is strongly related to whether or not the individuals are considering their own requirements or the requirements of others. Such prompts will not help the individual in an interactive setting generate requirements without careful consideration of group level phenomena. Developing a thorough set of requirements in an interactive group setting is a task which will require further exploration of group-level phenomena as well as other factors. The potential for superior group performance may be achieved if these phenomena are better understood in future research.

8. References


*Psychological Review*, **111**, 1036-1060.


Springer Boston.


Iivari, N. 2006. ‘Representing the user’in software development—a cultural analysis of usability work in the product development context. *Interacting with Computers, 18*, 635-664.


Miller, G. A. 1956. The magical number seven plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63*, 81-97


Reiter-Palmon, R., Herman, A. E. & Yammarino, F. J. 2008. Creativity and cognitive processes: Multi-level linkages between individual and team cognition. In:


Appendix A: BookExchange
Book Exchange Task Description

The Book Exchange is a website which will allow students at this university to buy and sell text books at a reasonable price. The website will not provide payment services; it will simply allow sellers to post items for sale, allowing potential buyers to search for their textbook offerings. The website will also have features that facilitate a buyer’s search for textbooks. For example, the website will have access to textbook requirements for a given course.

Your task:
- You will be directed in the completion of these tasks by the experimenter.
- Provide as many user stories as possible. A user stories is a story that provide a feature that the system to be designed should have. They are called stories because they tell a small story about how a user uses a feature of the website. So a feature must be something that describes the website itself. A recommended form for a user story is:

  “As a <type of user>, I want <some feature> so that <some reason>.”

For example:
“As a buyer, I want to be able to see the prices of all the books so that I can decide whether to buy the book or not.”

“As a seller I want to see if others are selling the same book so that I can determine my price accordingly.”

“As a professor I would like to have course book listings stored so that I don’t have to reenter the information each semester.”

“As a website administrator, I would like to be able to remove book listings so that I can clean up outdated listings.”

NOTE:
- Your stories should not be more than two sentences in length.
- You are NOT being asked to come up with a technical description of the website (i.e., it will use javascript pages). Instead we are asking you to describe what the system can do from the perspective of the users.
Appendix B: Requirements

1. User Account
   a. Account Setup (5)
      i. [Account] Create Account—general need for an account. If user stories mention an account, you should put it here, otherwise leave it in history.
      ii. [Personal Info] Store Personal Information—Change to edit and store personal information privately, to the extent that the user wishes
      iii. [Username/password]
      iv. [Forgot username/password]
      v. [School affiliation] Any features which address whether or not users should be associated with a specific school or schools

b. Book Advertising (12)
   i. [Post books]—general category
   ii. [Shipping Info] Shipping or Meet—details about the transfer of the book—must be actual information
   iii. Remove: List of books seller is selling These should all go in history. None of this section should be about being able to view what the seller is selling.
   iv. [Post multiple] books for sale The seller can post multiple books at a time for sale
   v. New: [Bundle books] The seller can bundle books together to sell them at the same time
   vi. [Provide steps]—Actual steps, nothing about easy to post books. If it mentions posting books, then put it in the posting book
   vii. [Seller Profile]—The seller can manage how he presents information about himself, including methods of communicating with him (email address, photo, style of the website, etc). This feature corresponds to what the buyer can see in Book information
   viii. [Post book info] Book info includes price and condition. Remember if it talks about posting, it goes here. If it talks about viewing, it goes in Book Condition under description
   ix. Preview pages of book
   x. [Upload Photo]—This could be about posting the actual picture of the book or a stock picture
   xi. [Preview ad]
xii. [Edit ad]—any time anything is edited by the seller. If the admin edits it, it should be under managing book ads

xiii. [Remove ad]—also this is only if the seller does the removing. If the admin does it, it should be in the managing book ads section.

c. Professor Capabilities (6)
   i. [Post course requirements]—Any time the professor inputs course requirements, as well as any comments about them
   ii. [Edit Course Requirements]
   iii. [Delete study Requirements]
   iv. [Designate Required/Optional]-This covers any features which separate required from recommended books for the course
   v. See required DO NOT USE – put in general
   vi. See optional DO NOT USE – put in general
   vii. New [Store book requirements]—This covers any feature which allows the professor to store textbook requirements for another semester
   viii. NOTE: Professors checking to see which books a student bought are out of scope.

2. Transactions
   a. Review (3) These features are about writing reviews, not about seeing them
      i. [Review Buyer]
      ii. [Review Seller]
   b. [Shopping cart] (1)—Any time a user mentions shopping cart or a means of saving multiple books to an order before the final purchase
      i. Add items
      ii. Clear list
   c. Buyer indicates intention to buy (3)
      i. [Book Change Notifications] Buyer can receive email updates about new books—This feature should cover any notifications about changes in the books available on the website—including whether or not new books are available or if the professor has made a change in the books
      ii. Auto-notified when course books become available—This feature will go away and be lumped in to the feature above
      iii. [Request books] that have or have not been posted—This feature addresses the buyer’s need to make requests for books that are already on the website or not. This could also be a wishlist.
      iv. Request books that have not yet been posted—this feature will go away and be lumped in to the feature above it
d. **[Seller Notification]** (1) This feature alerts the seller that someone has bought (or is interested in buying) one of his books. This is different from requests, since requests are for a particular book, not a seller’s particular book.

e. **History**— (5) This section is for viewing information on a buyer or seller’s profile related to the books they sell. If the existence or creation of a profile is mentioned, Create account should also be considered “hit” in Account Setup.

1. **[View Buyer’s own History]** Buyer can see own past transactions.
2. **[View Seller’s Own history]** Seller can see own past transactions.
3. **[View buyer’s History]** See selling history of buyers.
4. **[View Seller’s History]** See selling history of sellers.

f. **[Transaction Completion]**—This is any functionality where the seller indicates that a book is sold.

3. **Books**

a. **Book Content** (5)

1. **[Write book Review]** Buyer Post comments about the content of the book—Any user posts comments about a book.
2. **Seller Post comments about the content of the book**—buyer and seller post will collapse into the same feature.
3. **[See book Review]** View comments about the content of the book. These should mostly be about the book. If a buyer wants to know how often a book has been used in class etc. they would use a comment, so categorize under this feature.
4. **[Views]** See how many times a book has been viewed—this describes the seller’s ability to see how many people have looked at his or her individual ad. This does not describe views for the website or any other aggregate information. See track transactions under manage book ads.
5. **[See Requests]** See how many times a book has been requested.

This is not to see the requests for the seller’s individual book, but for the book with a certain ISBN.

b. **Book Condition** (3)

1. **[Condition description]**—See or know the condition of the book based off of a description, rating, etc.
2. **[See Actual Pictures]**—Specifically mentions pictures of the book. If it just says “see” the condition, use description.

c. **Book Price** (7)
i. **[Book Price] Price of the book**—This feature describes the price of the book, including other related information such as the price with taxes and shipping, the suggested retail price, etc.

ii. **[Buyback price] at bookstore**—self explanatory

iii. **[Bookstore price]** This feature reports the book’s price. If it doesn’t specifically mention bookstore, it shouldn’t go here

iv. **NEW: [Compare book price]**—This feature allows the BUYER to compare prices between the same book being sold by sorting by price or other means

v. **NEW: [External website comparison]**—This feature addresses comparisons with any external website or store, which enables the user to decide where they might buy the book

vi. **NEW: [Price Guide]**—This feature enables the seller to decide a correct price for the book by either looking at other book prices, using the website’s suggested price, or any other means.

d. **Book Information (10)**

   i. **[Description]**—Includes information about additional materials that could be included, but DOES NOT include additional materials. This also includes any oddball information not included in the other book information features including publisher, summary, size, keyword, etc.

   ii. **[Author]**

   iii. **[Edition]**—Edition can include anything that distinguishes a book from another book, including year, binding, but not ebook

   iv. **New—[ISBN]**—This has been moved from book browsing.

   v. **Title**—The title of the book—This is covered by ISBN

   vi. **Required or recommended [Course Requirements]**

   vii. **Professor**

   viii. **Parts of content of the book [Preview]**—This includes previews, the table of contents, etc.

   ix. **Picture of the book [stock picture]** Not a picture of the actual book. If the functionality is about finding the correct book, or specifies the cover, it should go here.

   x. **Contact info of the seller [Communication with seller info]**—This describes any functionality which enables the buyer and seller to communicate. Any communication between buyers or with the professor is out of scope

   xi. **[Seller Transaction information]**—The buyer can see transaction information (such as the meeting place, shipping method, shipping information) posted by the seller
e. Book Browsing (12)
i. General [Categories]—Use this feature to describe the ability to categorize, and any special method of categorizing that is not covered by the other features.

ii. Customize—Don’t use this category. It’s confusing and vague

iii. Classes [Course Requirements]—browse by a specific class within a course. This allows the student to see all courses which could correspond to a class. If the user wants to browse by professor, they can browse by the courses that the professor is teaching. There is no browse by professor category

iv. Prices REMOVE: Browsing by price is part of the price guide functionality and compare prices functionality.

v. [Subject]s—Subject can be area of study (e.g. psychology) or major, or genre.

vi. [Title]: If the seller or buyer wants to see how many books are left, they can browse by title to do this. We assume that title identifies a unique book

vii. Isbn REMOVE: There is no browsing by ISBN number (since we have title)

viii. [Edition]: Anything that distinguishes the same content—edition, volume, type etc.

ix. [Posting date]—when it was posted.

x. [Expiration date]

xi. [Author]—self explanatory

xii. New: [Condition]—Users can sort books by condition (E.g. new, used, etc)

xiii. New: [Course view]—this describes functionality where the user is presented with all courses at once, and the books that correspond to each course.

xiv. [Seller]—Any functionality about viewing a seller’s current listings of books goes here

f. Book Search (10)
i. General [search]—anything that mentions specific search functionality.

ii. [Subject] follow the guidelines for browsing features of the same name for the rest of these features

iii. [Condition]

iv. [Price]

v. [Title]
4. Administration
   a. Manage Book Ads (6)
      i. [Manage book ads]—any way of managing the books not covered, or is too general
      ii. [discover off topic] Discover books not required or suggested by professors—This feature should enable the admin to find any off topic books or other items being sold, or scams
      iii. [Discover out of date] postings—self explanatory
      iv. Discover duplicate postings—move these under manage book ads
      v. Discover new books—move these under manage book ads
      vi. Set a maximum price—move these under manage book ads
   b. Managing Users (9)
      i. [Manage Users]—general category
      ii. Web admin can [create account] for a user
      iii. Web admin can [delete account] for a user
      iv. Web admin can [edit account info]
      v. Web admin can [freezer user] account
      vi. Web admin can [ban user]
      vii. Web admin can flag innapropriate transaction—this should go under discover off topic in Manage Book Ads
      viii. Web admin can set maximum posting—move to general
      ix. Web admin can contact users—any communication from the admin to the user should be covered by this feature
      x. New: Track user transactions—Any functionality which includes aggregate user behavior for understanding demand for books, or otherwise improve the selling process, even if the seller requests it.
   c. Managing Reviews (3)
      i. [Edit] reviews
      ii. [Delete] reviews
d. System Support (5)
   i. **Comment section [feedback]**—This is any functionality which allows the user to give feedback about the website to the website owners or admin. It is not about books or about transactions with particular sellers
   ii. **[Help section]**—This is any material on the website which helps the user know how to use the website (aside from specific steps for posting books, which is in book advertising)
   iii. **FAQ: REMOVE**: This should be removed since anything here should go in “Help Section”
   iv. **NEW: [Contact support]**—This is any functionality where the user contacts the admin or other support personnel for help
   v. **NEW: [Policy]**—Any written material which explains policies for how to use the website, as well as privacy policies. This does not cover return policies. Returns are out of scope since the website does not sell books.
Appendix C: Coding Rules

Definitions:
User story: The subjects input. Consists of a user role, a feature of the system, and a reason for the feature that was stated.

Feature Concept: a concept mentioned in a user story that states or could imply functionality in the system. User stories may contain multiple feature concepts. The subject may have stated that the system performs something or may have stated them as an action that the user wants to accomplish.

High Level Feature: Lower level features are elaborations of the higher level feature. Are a summary of lower level features in a feature category and are designated by @@ symbols. They represent a summary of all lower level features and any features that could be part of the same summary of higher level features. If any lower level feature is hit, the high level feature is automatically counted.

Making implications: The following are guidelines on how to make implications:

- Use all information in the user story. For example, if a story about removing ads is framed from the administrator’s viewpoint, and a similar story exists for removing adds by the seller, you must choose the feature that relates to the administrator.

- Choose the most directly related features. Do not choose other features that must previously exist. For example, if a story suggests to have an account, but doesn’t specify whether or not the account needs a username and password, don’t mark the feature for account and password as hit.

- Different Direction: If a subject takes a specific feature concept in another direction (for example, subject says to use student ID and password to login, but our system says a user name and password is needed, there is a match between the subject’s contribution and our contribution).

- Use our feature to perform their feature (e.g. count of books—you can find this out by browsing, but it will take a while) is not allowed

- Principal for making implications: if we talk about that in the same amount of detail, take it.

Out of scope: The concept is not covered by our system’s features. It would not be considered as an elaboration of a higher level feature.
Appendix D: Informed Consent

**Adult Informed Consent**

An Investigation of Alternative Methods for Collecting Systems Requirements

**Invitation**
You are invited to participate in this research study. The information in this consent form is provided to help you decide whether to participate. If you have any questions, please do not hesitate to ask.

**Why are you being asked to be in this research study?**
The purpose of the study is to collect stories of the requirements of software used by individuals. You are eligible to participate because you are 19 years of age or older, or 18 years of age with parental consent, and are a student enrolled in courses at Creighton University.

**What will be done during this research study?**
During the 60 minutes that you participate in this study, be asked to provide information pertaining to the aspects and components you would look for in a software program and respond to a series of questions about your experience and demographic information.

**What are the possible risks of being in this research study?**
There are no known risks or discomforts associated with this study.

**What are the possible benefits to you?**
There are no direct benefits for participating in this study.

**What are the possible benefits to other people?**
This study may benefit society by contributing to our understanding of how to best develop computer software and programs.

**What are the alternatives to being in this research study?**
The alternative to participating in this study is non-participation.
What will being in this research study cost you?
There is no cost to you to be in this research study.

Will you be paid for being in this research study?
You will not be paid for being in this research study.

What are your financial obligations as a participant?
You have no financial obligations.

What compensation will you receive for participating?
You may receive class credit for participating in this research study. Your class professor determines the amount of class credit you will receive.

What should you do if you have a problem during this research study?
Your welfare is the major concern of every member of the research team. If you have a problem as a direct result of being in this study, you should immediately contact one of the personnel listed at the end of this consent form.

How will information about you be protected?
Reasonable steps will be taken to protect your privacy and the confidentiality of your study data. Any information obtained in connection with this study will remain confidential. The research data will be stored on a password-protected computer, accessible only by the primary investigator named above. No contact information will be obtained and no follow up responses to the survey will be pursued. The only persons who will have access to your research records are the study personnel, the Institutional Review Board (IRB), and any other person or agency required by law. The information from this study may be published in scientific journals or presented at scientific meetings but your identity will be kept strictly confidential.

What are your rights as a research subject?
You have rights as a research subject. These rights are explained in this consent form and in the *What Do I Need to Know Before Being in A Research Study?* and *The Rights of Research Subjects* available at: http://webmedia.unmc.edu/irb/RightsofResearchSubjects_English_091004.pdf.
If you have any questions concerning your rights or complaints about the research, talk to the investigator or contact the Institutional Review Board (IRB) by
What will happen if you decide not to participate?
Your participation in this study is voluntary and you may choose to withdraw at any time before, during, or after the research begins for any reason without any consequences. Deciding not to be in this research study or deciding to withdraw will not affect your relationship with the investigator, or with the University of Nebraska at Omaha. Your decision will not result in any loss of benefits to which you are entitled. If any new information develops during the course of this study that may affect your willingness to continue participating, you will be informed immediately.

If you have any questions during the study, you should talk to one of the investigators listed below. You will be given a copy of this consent form to keep. You are freely making a decision whether to be in this research study. Signing this form means that (1) you have read and understood this consent form, (2) you have had the consent form explained to you, (3) you have had your questions answered and (4) you have decided to be in the research study.

Signature of Subject: ____________________________ Date: __________
Time: ________

My signature certifies that all the elements of informed consent described on this consent form have been explained fully to the subject. In my judgment, the subject possesses the legal capacity to give informed consent to participate in this research and is voluntarily and knowingly giving informed consent to participate.

Signature of Investigator: ____________________________ Date:
________________________

Principal Investigator
Doug Derrick (402) 554-2060

Secondary Investigators
Joseph Allen (402) 280-3755
Aaron S. Read
Gert-Jan de Vreede, Ph.D.
Appendix E: Exploration of the Effectiveness of Individual Prompts

In another unpublished study, the time stamp feature of ThinkTank was exploited to identify which prompts were most effective at generating a more thorough set of user stories. ThinkTank associates a time stamp which identifies the year, month, day, hour, minute, and second that an item was contributed to its database by a user. This time stamp was hidden from the user during the experiments.

Using these time stamps and the timing of the prompts in the experiment, the quantity of contributions from each individual in the individual prompted and unprompted treatments following each prompt could be specified. The number of user stories contributing to a thorough set of user stories following each prompt can also be specified by using the categorization of the contribution as a user story using the original categorization made by one of the coders for the dissertation experiment. Duplicate user stories which occurred during the same prompting period were not counted. Figure 4 below shows the performance of prompted individuals at the time periods of 0-10 minutes, 10-17 minutes (following the first prompt), 17-24 minutes (following the second prompt), etc. The unique user stories generated in each time period were also identified for the unprompted groups. The comparison of the unique user stories generated by individuals in the unprompted treatment is depicted below in Figure 7.
The difference in the number of unique user stories generated by individuals in the prompted and unprompted treatment groups becomes more pronounced for the period following the prompt at 17 minutes when the top 10 performers in each group are compared (See Figure 5). Prompts may more effectively utilized by individuals with higher cognitive abilities (Valacich et al., 2006). This difference is depicted in Figure 8.

Finally, the ratio of features generated during each of the prompting periods is analyzed for the top 10 performers in each group. The difference between the prompted and unprompted feature ratios for the time period between 17 and 24 minutes was found to be
significant at the $p < 0.05$ level. Figure 9 depicts the differences in ratios for the two prompts.

Figure 9: Differences in Ratios of features generated for time periods following prompts