Easing Team Politics in Agile Usability

A Concept Mapping Approach

Jeremy T. Barksdale, Eric D. Ragan, D. Scott McCrickard
Center for Human Computer Interaction
Virginia Tech
Blacksburg, VA
{barksdale, ericragan, mccricks}@vt.edu

Abstract— **Team politics complicate software projects. They cause internal conflicts that can not only cost a software team time and money, but may also detract from the needs of the product's end users. In this paper, we explore the use of concept maps as a means of mitigating such team conflicts. Approaching agile usability through the lens of distributed cognition, concept mapping could improve team communications. We conducted interviews with eleven practitioners from three local software development companies to gain preliminary evidence of the practicality of the approach. Participants were questioned about their challenges in agile development and about their overall impressions of a concept mapping approach. We asked about the practicality and acceptability of implementing such a methodology, along with general concerns and recommendations. Results indicate that there is a need for improvement in agile usability, and a concept mapping approach is promising for addressing existing concerns. With refinement of the method and the development of the proper tools, this approach has great potential to improve team interaction in agile usability environments.**

Keywords-agile usability; distributed cognition; dcog; concept mapping

I. INTRODUCTION

Naturally, software project teams may experience tension at some points throughout the project lifecycle [15, 16, 19]. Members in various roles have different backgrounds and philosophies, which may lead to the formation of alliances within the team and/or dissention among team members. Although some degree of conflict may be considered healthy for debate, other forms of conflict can impede the development process and impact project cost and duration.

Agile development methodologies rely on team collaboration and communication to provide adaptive management and project organization [12]. These methods strive to identify problems and weaknesses within the project team before they grow to unreasonable levels. Collaboration in agile practices is vital in supporting the effective planning and management that makes it possible for teams to achieve their goals. While agile methods provide support for adaptive, accelerated development, they neglect sufficient consideration for usability [5]. To address this weakness, the agile usability approach incorporates usability engineering into the agile methodologies [16]. While establishing and maintaining strong collaboration among members is a major challenge for any agile team, the increased attention to usability adds further stress to the agile usability development system. Cultural and educational differences between software and usability engineers create communicational problems, and differing priorities can make it difficult for members to reach agreement. Because the software developers are responsible for the actual product implementation, the usability engineers often lose leverage in developmental decisions, which can harm usability. These issues are intensified when teams are geographically separated, as is common for many organizations.

In this paper, we are concerned with improving the interaction between usability engineers and software engineers on agile usability projects, mitigating cost and schedule overrun. With the aid of a lightweight approach, we believe project teams can work together to better accomplish the shared goal of satisfying the users.

Edwin Hutchins founded a distributed cognition framework, DCog [4], as a means to analyze cognition in social systems. DCog's unit of analysis is a system comprised of actors and artifacts, with internal processes of knowledge creation, representation, transformation, and propagation [4]. The representations and processes are key components in understanding how activities are distributed over several individuals and various media. Our approach utilizes DCog in that the team members are actors, the work products are artifacts, and their interactions with each other while accomplishing tasks are the processes. Our goal was to strategically improve the efficiency of this system by refining the process with the use of an artifact as an aid.

Founded by Joseph D. Novak in 1972, concept mapping was developed as a means of graphically organizing and representing children's conceptual understandings of science [18]. Recent use of concept mapping has proven useful not only as a learning tool, but also an evaluation tool.

Although concept mapping has been around for some time, its application to improving team interaction in an agile usability environment has not been found in the literature. We propose the use of concept maps as a novel approach to mitigate team conflicts. We argue that the recommended interaction modifications through the concept mapping approach will improve strategic weaknesses that significantly burden the current interaction in an agile usability environment while requiring minimal effort to apply—

978-0-7695-3768-9/09 $25.00 © 2009 IEEE
DOI 10.1109/AGILE.2009.57

resulting in a net gain. Through interviews with industry practitioners, we provide data to support this claim.

II. Related Work

In recent years, many have approached improving the interaction between software engineers and usability engineers. A general understanding of these techniques aids in conveying where our concept mapping approach fits within the landscape.

Lee [15, 16] offers an agile usability development process through applying extreme programming (XP) and scenario-based design to agile usability. Memmel [17] introduces a radical approach of integrating XP and Agile Modeling to form the CRUISER lifecycle which highlights a set of five phases and related deliverables. Göransson [11] argues that there should be a tight coupling between user-centered systems design and the software development process. Fox [8] also combines agile methods with user-centered design by providing a more linear process and describing the various approaches used to execute the roles of designers and developers.

We share with the above methods the goal of improving communication between the various disciplines. Our approach differs in that the above methods primarily focus on improving collaboration through improving process integration, where we focus on improving team interaction through the development of an artifact for integration within existing processes.

Earlier work shows evidence of DCog being applied to improve software team performance during perfective software maintenance [7]. In this empirical study, programmers were analyzed relative to each other, their tools and their environment as a complex cognitive system. DCog has also been referenced as a fundamental concept to Augur [9] – a visualization tool used to create visual representations of software artifacts and development activities in a distributed software environment.

No research was found on the use of DCog as a theory to support collaborative communication within agile usability project teams. Our goal was to begin addressing this gap.

III. Approach

In this section, we provide details regarding the approach. Specifically, we are interested in communicating a sufficient understanding of what the approach is, how it might be applied, and what its features are. We also provide a basic example of its use.

A. Agile Usability

Agile usability is concerned with bringing usability practices to agile development projects in order to address the weakness of agile projects not producing usable systems [1]. The agile methodology strives to increase the interactions between developers and customers and to deliver working software frequently, but is weak in delivering high usability [5]. To account for this weakness, agile usability integrates usability engineers into the team. The usability engineers employ usability instruments such as personas, user studies, and usability tests in order to strengthen their understanding of the users' needs [21]. The most common approach to agile usability applies the parallel operation of usability design and software development, but this requires efficient communication and coordination [15]. Major contributors Scott Ambler [1] and Larry Constantine [5] support the use of agile modeling to support the process. Agile usability has the potential for success because the developers and usability engineers share the common goal of providing software that satisfies the users' needs. Though they approach this goal from different perspectives, both sides want the user to have a product that they will want to use.

B. Distributed Cognition

Developed to address the need for an understanding of human cognition and system performance in real-world work environments, DCog emphasizes the social aspects of cognition and makes overall functional system performance the priority [4]. The framework involves analysis of information processing, problem solving, and knowledge transfer within the system. Knowledge is expressed through various representations across team members and tools, or cognitive artifacts, to aid in the storage and sharing of information [13]. This distribution of knowledge helps to support redundancy and communication of information throughout the system [14]. Successful coordination and communication of the distributed knowledge is vital for the success of the system, but establishing and maintaining these processes requires attention and effort [13]. Common application methods for distributed cognition include artifact analysis and social network analysis techniques [2]. The framework can be supported by context-aware tools that can increase resources for interpretation, prevention of information overload, and clarification of design intent [6].

C. Integration

Distributed cognition and agile usability are compatible approaches that can provide great value through integration. We believe that integration is a practical step that also lends itself to greater understanding of collaborative development.

Although agile methods often minimize documentation, external representations of knowledge are still essential for effective communication. Determining the scope of knowledge capture, understanding information transformations through the development process, and effectively communicating the necessary information to the stakeholders serve as the primary goals of distributed cognition. Agile usability requires this level of information sharing and communication. Thus, the distributed cognition framework provides a natural fit for supporting the improvement of agile usability.

The value of integrating distributed cognition and agile usability lies in the inherent social nature of the agile methodology and temporal constraints of agile projects. Distributed cognition has the ability to improve the throughput of the team and further support the agile usability goals of adaptability, accelerated delivery, and high developer-customer interaction. Successful integration will allow developers a methodology to help manage the

production of more usable products. Further, analysis of practicing systems and their informational artifacts may better the understanding of collaborative design.

D. Application

DCog-based agile usability uses an abbreviated version of Novak's concept maps [18]. High-level steps for usage include:

1) Identification of a design scenario by the usability engineers.

2) Publication of the scenarios by the usability engineers.

3) Creation of user stories and linking to their respective scenarios by the software engineers.

4) Collaborative annotation of the links between scenarios and stories.

5) Agreement on the concept map by the usability and software engineers and proceeding on to their respective processes.

The steps involved in creating a concept map fit nicely in the agile methodology as it itself is an agile process. For the purposes of distributed agile usability, concept maps will be used in conjunction with the Extreme Programming (XP) [3] approach and Nielsen's usability engineering process [20], which both emphasize the user's perspective as a top priority.

Though we use the XP approach here, this is only an example—this alignment is suitable for other instances of agile methodology. Nielsen's usability engineering process [20] begins in the "know the user" phase, where task analysis is conducted. The result of this task analysis is a description of the users' needs, which might resemble a scenario of their actions. As part of XP's primary practices, stories capture the customers' needs and determine what features will be implemented in the software system to provide "units of customer-visible functionality" [3]. As the usability and software engineers collaborate, the concept mapping process would begin in the overlap between "know the user" phase of the usability engineering process and the user story development of the software engineering.

Detailed concept mapping steps incorporate the following negotiation:

1) (Usability Engineer) Meet with users and create user scenarios of their day to day tasks or needs.

2) (Usability Engineer) Write up scenarios in their appropriate storage tool.

3) (Usability Engineer) Generate first version of the concept map providing the unique identifiers in the concept map.

4) (Software Engineer) Review the scenarios and generate a list of user stories based on the scenarios provided by the Usability Engineer.

5) (Software Engineer) Revise the concept map providing the unique identifiers of the stories and their relationships to the relevant scenarios.

6) (Software Engineer) Provide any necessary annotations as part of the relationships.

7) (Usability Engineer) Review the concept map, possibly with the user, and provide additional annotations as part of the relationships.

8) (Usability/Software Engineer) Continue negotiating on the relationships as needed.

9) (Usability/Software Engineer) Refine the concept map throughout the development process as needed.

E. A Basic Example

Following is an example of a user story derived from a design scenario, where a customer wants to withdraw money from an ATM.

Sample Design Scenario:

S1: "A customer withdrawing money from an automated teller machine (ATM). It's Friday afternoon and Joe is flying to Sydney. He doesn't have enough money for a taxi to the airport, and he's running late. He goes to the local ATM and identifies himself. He specifies that he wants $100 from his savings account. He'd like the money in $20 notes so that he can give the taxi driver the correct change. He doesn't want a printed receipt, as he doesn't bother keeping track of transactions in this account." [10]

Example User Story for Sample Design Scenario:

US1: Customers identify themselves using a bankcard.
US2: Customers enter the amount for withdrawal.
US3: Customers can select withdrawal denomination.
US4: Customers can choose whether a receipt is printed.

The concept map resulting from the connection of the scenario to the story has a parent node labeled "S1" and can have child nodes based on the user stories, "US1", "US2", "US3", and "US4." Connecting each of the child nodes to a parent node is an edge with text box in which the designers can describe the relationship. This annotation is a core feature of the approach, as it captures knowledge for reuse. Figure 1 serves as a simplistic example that might be created based on our example. The knowledge developed in the concept map may not necessarily be new information. Traditionally, this knowledge would be produced dynamically through conversation between usability and

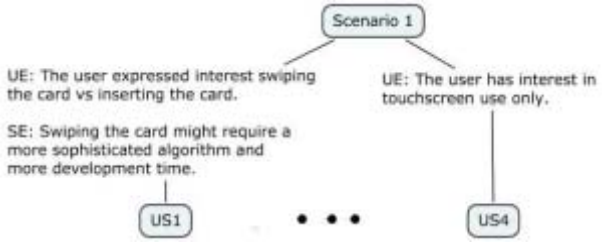

Figure 1. A simple concept map example showing the relationship between a scenario and user stories.

software engineers. While verbal conversation may be more convenient in the immediate term, it is inferior when the discussed information needs to be referenced later.

Note that in our sample design scenario, users are considered within the context of their worldly experiences; however, in the user story the user is situated relative to the system. Together, the scenario and story provide a complete picture of the user. This aligns the roles of the usability and software engineers, allowing the usability engineers to serve as the user representative and the software engineers to serves as the system representatives. The usability and software engineers should collaborate on the relationship between these user perspectives in order to effectively capture the user as a function of the system.

IV. METHODS

In this section, we provide information about the methods used to conduct this study. We provide a description of the participants, materials used, the procedures followed and what was measured.

A. Participants

Eleven companies, in the Blacksburg and Roanoke, Virginia vicinity were solicited for participation via email. Three small to mid-size companies responded and were selected for participation. Application domains of these companies include asset performance management, software engineering consulting, and digital services.

Across the three selected companies, we interviewed eleven software industry practitioners of varying roles. Specifically, nine participants were from one company and the remaining two were from two separate companies. The large number of participants from the one company was possible because a research lab colleague is an employee at that company. This provided us the ability to obtain a breadth as well as a depth perspective. The roles of the participants included one CEO, one Director of web application development, two Product Managers, three Software Engineers, one Development Lead, one Project Lead Manager, one Quality Assurance Intern, and one Development Manager. The mean number of years of participant experience, in their current role was 3.7 years, ranging from 1 to 16 years. The average total number of years of participant experience in the software industry was 12.2 years, ranging from 1 to 29 years.

B. Materials

In addition to an approved request for participation email and consent form, an approach overview document was used during this study for reference by the participant throughout the interview. Included in the document were high level steps, a step-by-step guide and a sample diagram to support the verbal description provided during the interview.

The questionnaire used to conduct the interview consisted of 22 questions. Of the 22 questions, seven addressed corporate and participant background. The remaining questions were relative to the approach on practicality, the participant's impression, acceptability, concerns, and recommendations.

C. Procedures

Although audio recording was covered in the consent form, verification was confirmed at the beginning of each interview. If participants did not get a chance to review the consent form when it was emailed, they were given the opportunity to review it at that time and were given an opportunity to ask any related questions. If the participant was comfortable continuing, the purpose of the interview, an overview of the overall research effort and the format of the remainder of the interview was given.

Participants were asked questions regarding the company's current practices and the participant's professional background. Following was a description of the proposed approach in preparation for the next set of questions. During the overview, a handout presenting the process steps was referenced to help clarify details of the approach. The participant was then given the handout for later reference after the participant communicated understanding of the approach. The next set of questions was core to the interview and specifically related to the participant's views on the proposed approach. Once the interview concluded, the description handout was collected and the participant was thanked for their time and energy.

D. Results

This section explains each of the categories of question types and lists the questions the participants were asked. An overview of the results (see Table 1) and summaries of the responses provided by the participants are included.

1) Practicality

Practicality is defined as the likelihood that the approach could be effectively used in agile development. It is an important factor in ensuring that the approach is one that would lend itself to practical use.

Would this approach be easy to implement?

- Five participants expressed that the proposed approach would be easy to implement.
- Five participants said that the proposed approach would be easy, contingent upon varying constraints or challenges being addressed. These challenges include how the approach implemented, the usability of supportive tools, and the team buy-in.
- One participant communicated that it would not be easy to implement because he did not believe the addition of documentation aligned with the agile methodology.

Would this approach take an unreasonable level of effort?

- Five participants expressed that this approach would not take an unreasonable level of effort to implement.
- Three participants communicated that the level of effort was based on contingencies, such as the ease of getting adjusted to the collaboration and the potential initial investment.

TABLE I. RESULTS OVERVIEW

Questions	Responses		
	Favorable	*Neutral*	*Unfavorable*
Practicality			
Would this approach be easy to implement?	10	0	1
Would this approach take an unreasonable level of effort?	8	0	3
Would this approach be difficult to learn?	11	0	0
Would this approach provide you increased job satisfaction?	7	3	1
How confident are you that this approach would fit in with your current practices?	8	0	3
Impression			
What are your thoughts about the use of this approach on your project?	7	1	3
Do you see potential benefits in the use of this approach? If so, what are a few?	11	0	0
Do you see potential costs in the use of this solution? If so, what are a few?	0	0	11
Do the potential benefit(s) will outweigh the potential cost(s)?	7	4	0
Do you see the process as a more strategic approach (addressing key team interaction issues) or brute force approach (merely another approach)?	11	0	0
Acceptability			
Would you encourage the use of this approach as part of your next project?	10	0	1
How likely is it that the other team members will adopt the use of this approach?	6	1	3
Concerns			
What do you dislike about the solution?	1	1	9

- Three participants said that this approach would take an unreasonable level of effort to implement due to the back and forth and demanding deadlines.

Would this approach be difficult to learn?

- All eleven participants communicated that this approach would not be difficult to learn. Two of the eleven, however, noted potential challenges of motivation and the need for clear communication of roles and expectations in the beginning.

Would this approach provide you increased job satisfaction?

- Seven participants responded that this approach would provide them increased job satisfaction due to reduced frustration from limiting the level of rehashing and improved ramp up time for new team members.
- Three participants expressed neutrality in that it would not affect their job satisfaction.
- One participant said that it would not provide increased job satisfaction because of the increased documentation.

How confident are you that this approach would fit in with your current practices?

- Eight participants communicated moderate to extreme confidence that this approach would fit in with their current practices since it was similar to their way of doing business.
- Three participants expressed little to no confidence that this approach would fit in with their current practices because of pressing client demands and the effort required to manage the map.

2) Impression

With the impression metric, the goal was to gain an understanding of the participant's overall perspective of the approach.

What are your thoughts about the use of this approach on your project?

- Seven participants expressed at least some level of interest in using this approach on a current or the most recent project. Some feedback along with these responses was that the approach would have to adapt and would be contingent upon team personalities.
- Three participants expressed that it might not be the best approach for a current project because the current project is mainly maintenance work.
- One person was too far removed from the most recent project and was unable to provide an answer to the question.

Do you see potential benefits in the use of this approach? If so, what are a few?

- All eleven participants noted that there were potential benefits to using this approach. A few of the benefits provided include faster ramp up, providing a pictorial representation, less rehashing of discussions, reference point, and reusability.

Do you see potential costs in the use of this solution? If so, what are a few?

- All eleven participants noted that there were potential costs with using this approach. Some of the costs were the time commitment, required initial effort, getting to consensus, and a policing cost to manage the use of the approach.

Do the potential benefit(s) will outweigh the potential cost(s)?

- Seven participants communicated that the potential benefits would outweigh the potential costs.
- Four participants expressed neutrality on whether the potential benefits would outweigh the potential costs.

Do you see the process as a more strategic approach (addressing key team interaction issues) or brute force approach (merely another approach)?

- All eleven participants communicated that this approach was more of a strategic approach than a brute force approach. This was based in the idea that it brought together existing processes and focused on collaboration.

3) Acceptability

We used acceptability to measure whether the participants thought the approach would be accepted by their colleagues. This was important in gaining perspective based on the participants' understandings of their team cultures.

Would you encourage the use of this approach as part of your next project?

- Eight participants said they would encourage the use of this approach on their next project.
- Two participants communicated that it was dependent on contingencies such as whether the tool is easier to use than employing current practices, whether there was team buy-in, and whether they would need a usability engineer on their team.
- One participant expressed that they would not encourage the use of this approach on their next project.

How likely is it that the other team members will adopt the use of this approach?

- Six participants said it was likely that other team members would adopt the use of this approach.
- One participant expressed uncertainty due to contingencies such as organizational culture and past team history.
- Three participants communicated that it was unlikely that other team members would adopt the use of this approach.
- One participant's response did not sufficiently address the question.

4) Concerns

Through inquiring about the participants' concerns, we hoped to understand what aspects of the approach might cause them to hesitate using the approach.

What do you dislike about the solution?

- One participant expressed that there was nothing that he disliked about the approach.
- Nine participants shared at least one dislike about the approach, including the lack of implementation details and increased complexity of the communication pattern.
- One participant's response did not address the question.

5) Recommendations

As a follow-up to learning the participants' concerns, we were interested in their ideas of how to mitigate those challenges as a starting point to refinement of the approach.

What modifications would you make to the approach?

- Two participants did not have any recommendations.
- Eight participants shared at least one recommendation, which included improving leverage of a tree data structure, add a quality assurance role, add dates to the concepts, and add guidelines of what is acceptable content to the process.
- One participant's response did not address the question.

V. Discussion

Three key themes were realized from analyzing participant responses. These responses included cost/benefit analysis, required time and effort, the need for more implementation details, and tool support. In this section, each of these will be addressed with the goal of adding clarity to how it relates to our hypothesis.

A. Cost/Benefit Analysis

Industry practitioners will often determine the usefulness of an approach or process based on whether the investment required is less than the return provided. As part of developing this approach, we strived to design the approach to align with such considerations by focusing on strategic weaknesses in team interaction.

Our results indicate, through a majority of positive responses on the cost/benefit ratio, that we have generally succeeded in accomplishing this goal. Furthermore, the fact that all favorable strategic responses might strengthen the claim that, in the long-term, the approach provides more value than the effort required. Collectively, these two questions directly support our hypothesis.

B. Time and Effort

When considering interaction refinement, dealing with change may require effort. This effort, in the short term, is acknowledged, but is expected to decrease throughout the use of the approach. We placed direct emphasis on designing the process such that it would take minimal effort. We indirectly focused on the time and effort by considering the approach's learnability, the ease with which it fits in with current practices, and its acceptability. We believed these factors to be important not only as a function of the cost/benefit ration, but also as way to alleviate frustration on projects.

The results show that participants generally view this approach as not taking an unreasonable level of effort, not difficult to learn, fitting in with current practices, and being

accepted by their colleagues. We take this to indicate that the aforementioned goals have, at least preliminarily, been addressed.

C. Implementation Details and Tool Support

Some participants expressed the need to see more details and noted that more information about the implementation is necessary to genuinely assess the approach. Through questions related to the overall impressions of the concept, our primary focus was on gaining perspective of practitioners' initial thoughts to guide our future direction. We saw value in taking this path as a way to increase the likelihood of satisfying practitioners' concerns before progressing too far down a potentially incorrect path.

Participant feedback and recommendations received from this study have provided useful insight about which factors should be addressed in moving forward with tool development. It will facilitate increased usability of the supporting tool as we are better able to address strategic needs.

VI. CONCLUSIONS

In this paper, we were interested in exploring whether industry practitioners would perceive the use of concept maps as a practical approach to improving team interaction on an agile usability project. We have provided background to the approach, specified the steps needed to apply this approach, and introduced concept maps as explicit rationale for representation transformations. Our findings indicate that participants found the approach to be potentially beneficial with some noted challenges and recommendations for refinement. We expect this approach to be an advantageous one. Agile usability has the problem of still being in its formative stages and lacking the aids to effectively manage team interaction. Our approach addresses this problem through the use of an artifact that facilitates better team communication. Our interviews confirm that this is promising. As next steps, we hope to conduct a case study to better understand the approach through use in a practical environment, develop a tool to facilitate that study, and refine the approach further based on the findings.

ACKNOWLEDGMENT

We thank Jason Chong Lee for his assistance with and guidance throughout conducting this research.

REFERENCES

[1] Ambler, S. Introduction to Agile Usability: User Experience Activities on Agile Development Projects.; Available from: http://www.agilemodeling.com/ essays/agileUsability.htm.

[2] Aranda, J. and S. Easterbrook. Distributed cognition in software engineering research: Can it be made to work? in Proceedings of Supporting the Social Side of Large Scale Software Development. 2006. Banff, Alberta, Canada.

[3] Beck, K. and C. Andres, Extreme Programming Explained: Embrace Change (2nd Edition). 2004: Addison-Wesley Professional.

[4] Carroll, J., HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science. 2003: Morgan Kaufmann.

[5] Constantine, L., Process Agility and Software Usability: Toward Lightweight Usage-Centered Design. Information Age, 2002. 8(8): p. 1-10.

[6] Fischer, G., et al. Supporting Collaboration and Distributed Cognition in Context-Aware Pervasive Computing Environments. 2004.

[7] Flor, N. and E. Hutchins. Analyzing distributed cognition in software teams: A case study of team programming during perfective software maintenance. 1991: Intellect Books.

[8] Fox, D., J. Sillito, and F. Maurer, Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfully Integrated in Industry, in Proceedings of the Agile 2008. 2008, IEEE Computer Society.

[9] Froehlich, J. and P. Dourish. Unifying artifacts and activities in a visual tool for distributed software development teams. 2004.

[10] Gaffney, G. What is a Scenario?; Available from: http://www.infodesign.com.au/usabilityresources/design/scenarios.asp.

[11] Goransson, B., J. Gulliksen, and I. Boivie, The usability design process-integrating user-centered systems design in the software development process. Software Process: Improvement and Practice, 2003. 8(2): p. 111–131.

[12] Highsmith, J. and A. Cockburn, Agile Software Development: The Business of Innovation. 2001.

[13] Hollan, J., E. Hutchins, and D. Kirsh, Distributed cognition: toward a new foundation for human-computer interaction research. ACM Transactions on Computer-Human Interaction (TOCHI), 2000. 7(2): p. 174-196.

[14] Hutchins, E. and T. Klausen, Distributed cognition in an airline cockpit. Cognition and Communication at Work, 1996: p. 15-34.

[15] Lee, J. and D. McCrickard. Towards Extreme (ly) Usable Software: Exploring Tensions Between Usability and Agile Software Development. 2007: IEEE Computer Society Washington, DC, USA.

[16] Lee, J.C., Embracing agile development of usable software systems, in CHI '06 extended abstracts on Human factors in computing systems. 2006, ACM: Montreal, Quebec, Canada.

[17] Memmel, T., F. Gundelsweiler, and H. Reiterer. Agile Human-Centered Software Engineering. 2007.

[18] Novak, J. and A. Cañas, The Theory Underlying Concept Maps and How to Construct Them. Florida Institute for Human and Machine Cognition, 2006. 1.

[19] Seffah, A. and E. Metzker, The obstacles and myths of usability and software engineering. 2004.

[20] Tucker, A.B., Computer Science Handbook, ed. A.B. Tucker. 2004: Chapman & Hall/CRC.

[21] Wolkerstorfer, P., et al., Probing an agile usability process, in CHI '08 extended abstracts on Human factors in computing systems. 2008, ACM: Florence, Italy.