UNIVERSITY OF CALGARY

An Environment for Collaborative Agile Planning

by

Lawrence Liu

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

Jan, 2006
UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "An Environment for Collaborative Agile Planning" submitted by Lawrence Liu in partial fulfilment of the requirements of the degree of Master of Science.

Supervisor, Dr. Frank Oliver Maurer, Department of Computer Science

Dr. Victoria Mitchell, Haskayne School of Business

Dr. Sheelagh Carpendale, Department of Computer Science

Date
Abstract

Existing project planning software for agile development processes offers limited support for face-to-face synchronous collaboration. This can be attributed to the common computer input paradigm – mouse and keyboard – in that they are designed for single users’ use. My proposed tool concept enables agile planning through the use of tablet PCs and tabletop displays to create a computerized environment that supports team collaboration. In this thesis, I designed and implemented a proof-of-concept system namely, AgilePlanner. It utilizes pen computing and digital tabletops technologies to emulate agile project planning using paper index cards and physical tables. AgilePlanner combines benefits from both physical and existing computerized planning solutions, and is built with an attempt to become an integral resource in the planning process. Results of the empirical evaluation with AgilePlanner reveal positive prospects of the tool concept as well as its shortcomings.
Acknowledgements

Reaching to the end of this research has been like chasing the sitting sun on the horizon; the ground to cover stretches further right before my eyes as I feel my fingertips are getting closer and closer to it. At times and I do mean *many* times, when this journey appears to me as nothing nearly as sufferable and unbearable to my past personal experience, it is only through the guidance, support, friendship and love am I able to make breakthrough to be where I am.

To Dr. Frank Maurer, my supervisor: I’d like to thank you for being the guiding light for the past two years. You always make me aware of dark corners surrounding my work and point out ways to get over them. Your critiques not only help me to become a more skilful researcher but also a more mature man. To Harpreeet: I am grateful for your making me realize how my egos usually set myself unrealistic goals, which sometimes paralyze me as I would be in constant concern of not being able to fulfill them. You help me tune down and get practical. I am also grateful for you as through our friendship you have shown me a life that is based on passion and love; which really warms my heart up, thanks. To Chris: I’d like to thank you for being as analytical and organizational as humanly can be. You are going to make some company very happy someday, I bet. Thank you for providing your invaluable feedback to my work time after time. To Wen and Nacy: I’d like to thank you for making our life under the same roof comfortable and memorable. Through interaction, you make me realized certain personal areas that I need to grow and the improvement also benefits my research. To Kris and Thomas: I thank you for providing pointers to help my research out of your own research experiences. To
Grigori and Carmen: I thank you for being ever so energetic and cheerful. You really light up this research group.

To Ryan: I thank you for being there when I needed someone to listen to the most. Thank you for always trying to cheer me up and encourage me to go on with my research and life. To Jason: I thank you for proofreading my thesis and making it better.

To my family: I thank you for always supporting me and being there taking care of me. I thank my parents for exemplifying what “never give up” means, and I thank my older brother for showing me his personal transformation and prove to me that people can change if they really commit themselves to their causes.
Dedication

To my family, who are always willing to support me through thick or thin.
# Table of Contents

Approval Page ......................................................................................................................... ii
Abstract ................................................................................................................................. iii
Acknowledgements ............................................................................................................... iv
Dedication ............................................................................................................................... vi
Table of Contents .................................................................................................................. vii
List of Tables ......................................................................................................................... xii
List of Figures ......................................................................................................................... xiii
List of Symbols, Abbreviations and Nomenclature ................................................................ xv

CHAPTER 1. INTRODUCTION ............................................................................................... 1

1.1 Planning in Software Project Management ................................................................. 1

1.2 Planning in Agile Project Management ........................................................................ 3

1.3 Agile Iteration Planning and Existing Planning Tools .................................................. 4

1.4 Ubiquitous Computing .................................................................................................. 6

1.5 Research Motivation ..................................................................................................... 7

1.6 Thesis Problems .......................................................................................................... 8

1.7 Thesis Goals ................................................................................................................ 8

1.8 Structure of Dissertation ............................................................................................ 10

CHAPTER 2. RELATED WORK .......................................................................................... 12

2.1 Importance of Planning in Software Project Management ........................................... 12

2.2 Agile Project Management and Agile Iteration Planning Process ................................ 14

2.3 Existing Tool Support for Agile Planning ..................................................................... 17
5.1 Supported Criteria ........................................................................................................ 80
5.2 Limitations .................................................................................................................. 83
5.3 Summary .................................................................................................................... 85

CHAPTER 6. EMPIRICAL EVALUATION ........................................................................ 86
6.1 Objectives .................................................................................................................. 86
6.2 Experimental Context & Participants ........................................................................ 87
6.3 Instruments ................................................................................................................ 89
6.4 Data Collection and Analysis ..................................................................................... 89
6.4.1 Evaluation Criteria ............................................................................................... 89
6.5 Interview .................................................................................................................. 92
6.6 Case Study ............................................................................................................... 92
6.6.1 Step One: Original Planning Practices ............................................................... 93
6.6.2 Observations ....................................................................................................... 94
6.6.3 Step Two: Planning Practices using AgilePlanner ............................................. 96
6.6.4 Observations ....................................................................................................... 98
6.6.5 Interpretation ....................................................................................................... 100
6.7 Limitation ................................................................................................................ 106
6.8 Summary ................................................................................................................ 107

CHAPTER 7. CONCLUSION AND FUTURE WORK ................................................. 108
7.1 Thesis Problems ...................................................................................................... 108
7.2 Thesis Contributions .............................................................................................. 109
7.3 Future Work ........................................................................................................... 111

REFERENCES .............................................................................................................. 113
APPENDIX A: INTERVIEW TRANSCRIPTIONS................................................. 119
APPENDIX B: ETHICS APPROVAL.......................................................... 133
APPENDIX C: ETHICS APPLICATION...................................................... 135
D.1 Consent Form.................................................................................. 135
List of Tables

Table 3.1: Assessment for existing tool support for agile planning. .......................... 52
Table 5.1: Assessment of AgilePlanner by the 17 tool support criteria in comparison with
existent planning too ........................................................................................................... 79
Table 6.1: Summary for usefulness comparison between AgilePlanner and the team’s
original planning tools ........................................................................................................ 100
List of Figures

Figure 1.1: The research context and scope of this work ................................. 10

Figure 2.1: Interaction diagrams among software development processes identified by
PMBOK; image from [Duncan and PMI Standards Committee 1996] .................. 14

Figure 2.2: VersionOne: an example of form-based system interface .................. 22

Figure 2.3: Editing Wiki text using a textbox of standard Web browsers .............. 23

Figure 2.4: The rendered Wiki page .................................................................... 24

Figure 2.5: MASE allows structured (a) and unstructured (b) information to co-exist on
one page. .............................................................................................................. 25

Figure 2.6: Glue Wiki represents iteration as a planning board, and story cards to can be
placed on the board as a metaphor that they belong to such iteration ............... 26

Figure 2.7: the iLab table ....................................................................................... 36

Figure 2.8: Illustration of RNT motion – image extracted from [Kruger et al. 2004] ..... 41

Figure 4.1: AgilePlanner – a computer-aided environment for iteration planning .... 60

Figure 4.2: An agile team using AgilePlanner to conduct iteration planning .......... 61

Figure 4.3: Creating a story using AgilePlanner and a Tablet PC .......................... 62

Figure 4.4: User interface of the story editor in comparison to paper index card in human
handwriting ............................................................................................................. 63

Figure 4.5: The view of AgilePlanner on the tabletop before (left) and after (right)
systematically reordering ..................................................................................... 64

Figure 4.6: Bug, refactor, and feature stories from left to right .............................. 65
Figure 4.7: On Tablet PCs, AgilePlanner shows all the iteration selected for current planning and their children stories in a tree structure. .................................................. 66

Figure 4.8: The left and right images contain the same set of story cards. The image cards in the green area are smaller than cards out of it by comparison. As stories are placed into the iteration, managerial metrics are updated demonstrated by (a) and (b). The round-shaped areas, free rotation of the image cards and the shirking effect on the cards inside iteration is produced by the iLab table software. ......................... 68

Figure 4.9: MASE Web interface displays agile plan information...................................... 69

Figure 4.10: The architecture of AgilePlanner Client Core (left) and AgilePlanner (right), as well as a demonstration of a typical execution sequence of the system.............. 73

Figure 5.1: Agile planners engage in collaboration using a corner of the iLab table and a Tablet PC. .......................................................................................................................... 83

Figure 6.1: Pictorial representation of the physical setting of the first experimental step.93

Figure 6.2: The real planning setting for the first experiment step .................................. 94

Figure 6.3: Pictorial representation of the physical setting of the second experimental step .......................................................................................................................... 97

Figure 6.4: The real planning setting for the second experimental step ......................... 97
List of Symbols, Abbreviations and Nomenclature

UC technology: Ubiquitous Computing technology
HCl: Human Computer Interaction
SWEBOK: Software Engineering Body of Knowledge
PMBOK: Project Management Body of Knowledge
PenC technology: Pen Computing technology
PMI: Project Management Institute
XP: eXtreme Programming
CHAPTER 1. INTRODUCTION

This thesis investigates the feasibility of combining the benefits of paper- and electronic-based media to support agile project planning processes. I begin this chapter with a discussion of the importance of project management – specifically, project planning processes – in the realm of software engineering. Next, I present agile project planning, and point out the limitations of existing planning tools. In light of these limitations, I present the proposed solution and the specific goals of this research. Finally, I conclude this chapter with a structural overview of this thesis.

1.1 Planning in Software Project Management

Software engineering is defined as “the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software” [IEEE 1990]. It encompass a set of three key elements that enable a project manager to control the process of software development – methods, tools, and procedures [Pressman 1994]. Methods provide the technical ‘how tos’ for building software [Pressman 1994]. Tools empower software engineers to concentrate on the creative aspects of software lifecycle process by reducing their cognitive loads [IEEE 2004]. Procedures define the sequence in which methods will be applied, and when the deliverables (project plans, documents, reports, etc.) should be generated [Pressman 1994]. These deliverables are usually generated using tools. The Project Management Institute (PMI) describes project management as “the application of knowledge, skills, tools, and techniques, to project activities in order to meet or exceed stakeholder needs and expectations” [Duncan and PMI Standards Committee 1996]. Project management practices are usually prescribed in
software engineering procedures to control and coordinate changes while ensuring
software quality, to assert milestones that allow software managers to assess project
progress, and to manage project resources to produce software in a productive manner.
These goals are primarily achieved by the means of project plans. To produce a good
project plan for software development, the planners should be aware of the unique
challenges that software projects bear [Griffiths 2004]. Firsts, software is intangible and
difficult to explain well; this contributes to mismatches between interpretations of
original requirements and customer goals. Secondly, the process of executing a software
project plan is a complex and high risk activity; detailed task-oriented plans are likely to
lead to fragile, soon abandoned plans. Finally, software has “extreme modifiability” that
allows changes to be made late into the project. Coupled with the inherent problem of
getting requirements right the first time, software projects are more likely to drift away
from their original specifications. Non-agile software project management drew heavily
on the principles from engineering that stress predictability [Augustine and Woodcock
2003]. “Predictive planning” [Larman 2004] is inherently and intensively demanding on
stabilities – stable requirements, analysis, and stable design [Augustine and Woodcock
2003]. This rigidity can cause project planning to diverge from reality into an illusion
[Beck 1999]. This illusion can cause the developers to get frustrated watching their work
down to the drain, and the customers to become angry as they have been made promises
based on that illusion.
1.2 Planning in Agile Project Management

“While managers designed traditional methodologies in an effort to control projects, the technical community gave birth to agile methodologies in response to their frustrations with traditional management and the resulting impact on their products and morale” [Augustine and Woodcock 2003]. Put equivalently, Agile Methods [Abrahansson et al. 2002] differ from traditional software development methodologies to a large extent by the practices of project management. Agile projects embrace changes and attempt to minimize the consequent risks by employing iterative software development [WIKIPEDIA: Agile software development 2005]. Each iteration goes through a complete mini-lifecycle of software development – planning, requirement analysis, design, coding, testing, and documentation – to deliver incremental functionality and a working system [WIKIPEDIA: Agile software development 2005]. Agile projects emphasize ongoing face-to-face communication within the teams including customers. In addition, agile projects emphasize creating working software over documentation.

Planning in agile project management – agile planning – is evolutionary and adaptive. “Adaptive planning” [Larman 2004] implies there is no detailed plan for all future iterations and releases; no fixed number of them, how long they are, and what will be done in them. Milestones to the project are left flexible. Adaptive planning is based on the belief that the precision is commensurate with the information; to plan in detail only up to some realistic planning horizon, beyond which things become speculative [Larman 2004]. Adaptive planning is superior to predictive planning in terms of working toward milestones, because each planning step can be done based on most up-to-date information
[Larman 2004]. Moreover, adaptive planning not only can reduce risks but also can adapt to emerging opportunities [Larman 2004].

According to the most exercised agile methods, XP (eXtreme Progamming [Beck 1999]) and Scrum [Larman 2004], agile projects include pre-project, release and iteration planning. The aim of the pre-project planning is to assess the feasibility of the project. The outcome artefacts of these three types of planning are a first plan, release plans and iteration plans. A release plan consists of stories extracted from the first plan. These stories may be split into finer-granulated tasks and distributed among the iterations under a release. The difference between the release and iteration plan is that “the release plan is synchronized to the rhythms of business. It gives the business people a way of thinking about sets of stories that together tell a good story to the market. The iteration plan is synchronized to the rhythms of programming” [Beck 1999]. These three types of agile project plans aim to resolve management issues on different scales. Yet, in terms of planning tools, pre-game, release, and iteration planning all employ paper-index cards and optionally project management systems.

1.3 Agile Iteration Planning and Existing Planning Tools

Agile Methods are iterative and evolutionary. Being iterative means the methods segments the lifecycle of software development in multiple time frames; in each time frame a small part of a system is completed and some business values gets delivered to the customers. Being evolutionary means product development is driven by customer feedback. [Larman 2004] identified twelve key motivations and benefits that result from using iterative and evolutionary development. Overall it is about better system
complexity and risk management, instant feedback towards the team’s confidence and
customer satisfaction building, and regular and early process improvements.

Agile iterations begin with planning. An iteration planning meeting is attended by
customers, developers, project managers and possibly other stakeholders of the project.
The meeting is usually held in a collocated manner to take advantage of the
communicational richness that direct face-to-face interaction provides. The customers are
supposed to identify and prioritize new user stories for the current iteration. The
developers are responsible for assessing feasibility and time estimates for each story
based on their expertise. The planning process consists of the following types of activities.
During the first type of activities, user stories are created and edited. Such activities are
usually performed by individuals alone. The second type of activity is concerned with
story organization, prioritization, and the discussions that go along. These activities, as
opposed to the first type of activities, are performed by all members of the planning party.

Existing planning tools are of two types. One is physical, and the other is computer-
aided. It has been observed that physical planning media are more used, and in general
considered more effective than computer-aided tools during planning. Detailed
discussions about these two types of planning tools and three resultant planning practices
are provided in Chapter 2. In a nutshell, modern computers are normally designed for the
use of single users. Interacting with these computers during planning tends to divert
planners’ attention away from collaboration and leads them towards tool usage instead.
On the other hand, physical planning media, although more intuitive to work with during
collective activities, do not support effective and instant plan dissemination, thus are of
little use to distributed software development. Furthermore, as it is also discussed in
Chapter 2, the outcomes of agile planning are supposed to be used for various activities in development. Information persisted on physical media lacks of the capability of being manipulated automatically, systematically and dynamically. As such, there exists a need for a planning tool that can offer the aggregated benefits from both types of agile planning tools.

1.4 Ubiquitous Computing

The concept referred to as Ubiquitous Computing (UC) promotes embodiment of computer assistance into the background of problem solving activities so that people’s focus can return back to tasks again [Weiser 1991]. UC is driven from a fact that traditional computers largely live in a world of their own – dealing with them usually means handling complex jargon that usually has nothing to do with the real tasks [Weiser 1991].

As an example, Weiser and his colleague [Weiser 1991] have built what they have called tabs – inch-scale machines that are approximately the sizes of Post-it notes and function as active badges. These badges identify themselves to receivers placed throughout a building so that it is possible to keep track of the people or objects to which they are attached. The tabs can be interconnected. As well it can be extended to work with other inch-scale computers, and augment functionalities of existing computing appliances.

UC can be integrated into an agile planning environment to allow planning participants to work with a physical-planning-media-like interface while realizing the
benefits of electronic medium such as, real-time data persistency and distribution, remote and distributed information access, dynamic computation powers, etc.

Two UC technologies can be utilized to support agile planning are:

- Digital tabletop technologies. Digital tabletops can be used to hold and organize project plan information.
- Pen computing technologies. Pen computing enabled devices can be used to create and enter project plan information.

Details of how these two technologies are used in my research will be manifested in the following chapters as this thesis work unfolds.

1.5 Research Motivation

At present, tool support to agile planning activities is rather limited; first, existing project management systems do not support physical-planning-like interfaces. These systems are usually designed for the use of single-users; which cannot be planned on concurrently and collaboratively. Secondly, in distributed agile planning agile practitioners are forced to use these existing systems as physical planning media are of little use. Ultimately, this dilemma may be resolved by enabling agile practitioners to apply traditional planning techniques to network-connected planning meetings. My research undertakes the first step toward such a direction to explore the feasibility of combining the benefits of paper- and electronic-based agile planning tools into a computer-embedded environment that facilitates co-located as well as distributed agile teams.

UC is an advancing Human Computer Interaction (HCI) field. Most digital-tabletop-based applications are research prototypes focusing on enabling human-table interactions
in a general context. No tabletop applications have I come across in my research are
designed to support a specific work process such as agile planning in software
development.

My research is motivated, first of all, to investigate the short-term goal mentioned
above. Secondly, since digital tabletops remain a novel technology and their applications
are mostly designed to enable traditional table interactions, it would be interesting to
investigate if the concept can support agile planning processes. As previously explained,
agile planning can be on various scales, yet they all employ similar planning
methodologies regarding tools and their usage. As such, although my target is agile
iteration planning in this thesis, the effects of this research can be applied to all types of
agile planning. Therefore, I will use the terms agile and iteration planning
interchangeably in this thesis.

1.6 Thesis Problems

In this thesis I set out to address the following problems:

1. It is unknown what the requirements should be for a tabletop-based computer tool
to facilitate collocated face-to-face collaboration for agile teams during iteration
planning;

2. If such a tool support is built, it is unknown how useful and usable it will be
compared to existing planning tools.

1.7 Thesis Goals

In this thesis, I will address the aforementioned problems with the following goals:
1. I will investigate the effects of facilitating iteration planning given existing planning tool support;
2. I will identify a set of requirements for tabletop-based computer support that are essential to assist activities during and after the iteration planning process;
3. I will design and implement a proof-of-concept tool that facilitates the iteration planning process according to the proposed approach; and,
4. I will empirically evaluate the effectiveness of the proof-of-concept tool.

Figure 1.1 summarizes the context and scope of this research. Software engineering is a professional area that creates and maintains software applications by applying technologies and practices from many fields including project management. This research focuses on a sub-area overlapping the fields of project planning from project management and agile methods from software engineering. Planning sets milestones for a project so that the progress can be measured, and coordinates different processes that are marching in parallel toward a common goal. As agile methods understand the unique nature of software projects, they prescribe adaptive and iterative planning as a core process to software. My research concentrates on investigation of planning issues of software teams that practice agile methods, and the associated issues in tool support.
1.8 Structure of Dissertation

The remainder of this thesis is organized as follows:

Chapter 2 details agile planning processes, and presents the usage of existing agile planning tools and their user perceptions. This chapter also provides descriptions of the UC concept given the context of this research, and demonstrates the perceived benefits to agile planning through surveying UC applications.

Chapter 3 extracts a list of requirements for tabletop-based computer support to facilitate face-to-face interaction during agile planning, and the project management activities that follows. They become essential tool criteria on how embedded computer support can assist iteration planning process as a whole. Also illustrated is the current state-of-the-art of existing planning tools in fulfilling these criteria.
Chapter 4 presents the design and implementation of a proof-of-concept tool, AgilePlanner, in detail.

Chapter 5 provides a qualitative analysis comparing AgilePlanner against the tool criteria elicited in Chapter 3.

Chapter 6 provides an exploratory evaluation of AgilePlanner. The aim of this research experiment is to investigate the usefulness and usability of AgilePlanner and to shed light on the potential impact of the proposed tool concept.

Chapter 7 concludes this research and suggests possible future improvements in the research area related or similar to the proposed tool concept.
CHAPTER 2. RELATED WORK

In this chapter, I present the existing literature relevant to both agile planning and UC. I start with a discussion regarding the significance of planning processes from a software project management perspective. I then zoom in onto agile project management with a specific focus on iteration planning, and review strengths and weaknesses of existing agile planning tools. My proposed tool concept is to incorporate UC into agile planning. I provide an existing UC application as an example to further illustrate the potential benefits from such concept. This research employs two sub-UC technologies – digital tabletop display and Pen Computing (PenC) technologies. As a result, a customized digital table, and pen computing enabled devices are acquired for my research. The integrated use of these two types of devices is similar to an implementation model that supports collocated collaboration, Single Display Groupware (SDG), which is presented at the end of this chapter.

2.1 Importance of Planning in Software Project Management

In general, software projects need to be planned out because

- “We need to ensure that we are always working on the most important thing we need to do” [Beck 1999]. Without planning, we might spend efforts on things that are not most important. This induces “opportunity cost” [WIKIPEDIA: Opportunity cost 2005] – time spent on doing one thing could have been spent on something more important.
• “We need to coordinate with other people” [Beck 1999]. Software is full of coordination: marketing, financial periods, management promises, etc [Beck 1999]. Planning allows us to get an idea what is feasible and reasonable.

• “When unexpected events occur we need to understand the consequences for the previous two” [Beck 1999]. Planning helps us to consider the effects of the unexpected effect and quickly adjust to these effects as soon as we know them.

The project management body of knowledge (PMBOK) [Duncan and PMI Standards Committee 1996], developed by the Project Management Institute, provides guidelines for managing projects in general, including software projects. The guidelines outline nine managing areas, and recommend managing processes and techniques for each of them, including: integration, scope, time, cost, quality, human resource, communication, risk and procurement. A general framework of five process groups are identified are common management techniques, including initiating, planning, controlling, executing, and closing processes [Duncan and PMI Standards Committee 1996]. PMBOK defines and depicts the interaction among the three major process groups as follows:

• **Planning processes**: this type of processes involves devising and maintaining a workable scheme to accomplish the business needs that the projects are to address;

• **Executing processes**: this type of processes concerns coordinating people and other resources to carry out the plan; and

• **Controlling processes**: this type of processes focuses on ensuring that project objectives are met by monitoring and measuring progress and taking corrective actions when necessary.
Figure 2.1: Interaction diagrams among software development processes identified by PMBOK; image from [Duncan and PMI Standards Committee 1996].

Software planning is important as its outcomes drive the rest of management endeavours. The circular interactions among the three major processes dictate an intrinsic requirement for planning artefacts – that is the outcomes of software planning need to be created, edited, maintained, extracted and manipulated with a great deal of flexibility in how they can be accessed and stored.

2.2 Agile Project Management and Agile Iteration Planning Process

Agile project management implements time-boxed iterative development. Being iterative means the development lifecycle is segmented into multiple iterations, and system functionalities are incrementally released iteration after iteration until a system is completed as a whole. Iteration time-boxing is the practice of fixing the iteration end date and not allowing it to change. When the goal of the iteration cannot be met, the scope is reduced rather than slipping the end date. [Larman 2004] pointed out twelve motivations
that promote the adoption of iterative development for agile projects. These twelve motivations are equally important. Here I randomly picked four of them to illustrate the benefits of such a development.

- Research shows iteration development is associated with lower risk and better success. These results have led large and experienced software procurement organizations such as the USA Department of Defense to promote the use of iterative development.

- Risk-driven iterative development encourages the teams to tackle the hardest, riskiest problems first. It drives the truth of risk to emerge, and forces unexpected issues into the open.

- Short iterations lead to a quick and repeating sense of completion, competence, and closure. The psychological factors are important for individual satisfactions, team confidence, and customer confidence in the team as they see steady progress in the direction they care about.

- A common practice of iterative development methods is a pre-iteration assessment – for example, a fifteen minutes discussion to discover concrete actions to improve the living process. Thus, iterative development leads to early and regular process improvement.

The outcomes of agile planning are stacks of index cards describing user stories. The stories should have business values and be understandable to customers as well as independent of each other. They are the units of functionality, which can be used to monitor and demonstrate the project progress. These stories serve the execution processes of iterations as cues that remind developers of unfinished tasks before deadlines, and
trigger communication with customers if further clarifications are needed. In terms of the controlling processes during iterations, project managers would check the daily updates of the stories from the developers, and determine the overall status of the project.

Agile iteration planning is usually conducted around a large horizontal surface i.e. a table. I assume a horizontal tabletop to represent a shared planning workspace for the following discussions. The planning process can be thought of as consisting of two main activities. During the first main activity, user stories are created. The stories serve as a precursor for future conversations between the customer and the developers. The content of the stories and the ensuing discussions during the planning meeting should be just detailed enough to estimate the implementation effort for the stories. Paper index cards are usually chosen to record the user stories because of their convenience. The limited space offered by the index cards force the story content to be concise. After creation, stories are usually placed in an area representing the Product Backlog (an artefact used in the Scrum method [Schwaber 2004] that stores major items of remaining work) to be processed later, or laid out on the tabletop to be revised, organized, and processed immediately. During the second main planning activity, team members together process the created stories: they reorganize, revise, categorize, split, estimate the development effort, and prioritize them. Different physical areas of the planning space often represent different project stages. For example, the pile of story cards that constitute the Product Backlog may be stored in one corner of the tabletop (the future). The central area may be used to organize the stories to be completed in the current iteration (the present).
2.3 Existing Tool Support for Agile Planning

Existing tool support for agile planning can be categorized into two types: physical and computer-aided. Physical planning media involve pen, paper index cards, tables, whiteboards, etc. Computer-aided planning media refer to existing agile project management systems, such as VersionOne [VersionOne 2005], MASE [E-Business Engineering Group 2005], Rally [RallySoft 2004], etc. There are three planning approaches associated with these two types of planning media.

**Approach 1.** Using physical planning media to support activities during and after iteration planning (Physical Media Planning).

**Approach 2.** Using existing agile project management systems to support activities during and after iteration planning (Computer Media Planning).

**Approach 3.** Using physical planning media for supporting activities during the planning meeting and existing agile project management tools after iteration planning (Combined Media Planning).

2.3.1 Physical Planning Media

Physical planning media involve pen, paper index cards, tables, whiteboards, etc. In this research I assume tables as the working area during planning because it is commonly used in face-to-face collaboration setting according to my own experiences. Advantages of physical planning media can be broken down into those that involve the use of paper index cards and those that involve the use of physical tables.
2.3.2 Physical Media Planning – Advantages

The advantage of paper index cards is that they can be used to capture random ideas [Cave 2003]. Pirsig [Pirsig 1991] described how a system of index cards helps to record randomly emerging thoughts as following:

“They (index cards) ensured that by keeping his head empty and keeping sequential formatting to a minimum, no fresh new unexplored ideas would be forgotten or shut out.”

“When information is organized in small chunks that can be accessed and sequenced at random it becomes much more valuable than if you have to take it in serial form.”

“Periods started to appear when he just sat there for hours and no slips came in – and this, he saw, was at last the time for organizing.”

According to Pirsig, the use of paper index cards imposes a minimal formality which allows for easy adoption by human beings in brainstorming activities, such as planning. The common size of an index card is 3×5” (76x127mm). The limited space is perceived by some as a benefit because it helps them focus on the core of tasks. When a task is too large, the size of cards serve as a physical reminder prompting users to break the task down into smaller ones. Creating, editing or discarding the index cards is lightweight and they are easy to pass around and share.

Kruger [Kruger 2004] highlights seven characteristics of traditional tables that mediate collaborative activities in his M.Sc. thesis. The following explanation of these characteristics is reworded from his work:
1. **Reach.** Typically people can reach any area on a table’s surface. That means the working area is as large as the table surface.

2. **Seating.** People can sit down when engaging in interaction around a table. Benefits to collaboration are three folds. First, seating allows planning participants to sustain a longer time of collaboration. Second, seating results in more comfortable user arrangement. In addition, hardly any one switch chairs during the course of collaboration, planning around a table helps people locate each other.

3. **Engagement.** With tables, people can easily adjust their distance to each other and how directly they want to face each other. It is an important comfort factor for adult collaborators since planning around tables allows the user to maintain their personal territories [Hall 1966].

4. **Personal space.** Personal space is the work area directly in front of each individual by convention. No need to appoint an individual area when personal work needs to be undertaken during collaboration.

5. **Holding of objects.** People can place arbitrary objects on a table’s surface; which is likely a key reason groups gather around tables – it is convenient and easy.

6. **Sharing.** People can easily share artefacts by passing them around a table.

7. **Casual interaction.** Because of flexible seating arrangements and easy reaching of objects on a table’s surface, a table encourages casual interactions.

According to my observations, these collaborative characteristics of traditional tables make them favourable for agile planning.
In sum, the use of paper index cards facilitates users to capture emerging ideas. They are effective medium for communicating and sharing. The use of physical tables encourages face-to-face interaction, and allows for easy editing and organization of the paper cards. As such physical planning media are in general considered the most effective tools for planning collaboration.

2.3.3 Physical Media Planning – Disadvantages

The use of physical planning media induces challenges to planning activities. First, bookkeeping of paper index cards is difficult. Managing lost and misplaced cards can be a recurring battle facing a development team.

Second, it is hard to maintain a complete history of user stories with paper index cards. A complete story history that provides raw data for requirement evolution analysis, is difficult to achieve using paper index cards simply because the work is too cumbersome and laborious.

Third, it is infeasible to distribute the agile planning meeting with paper index cards. One cannot physically keep all the physical cards at one central location while allowing others to have remote access to them. Card duplication might be in order in that scenario; however, replicated data can be out of sync with each other; which results in confusions.

Fourth, attaching external resources to the paper index cards is difficult for example, a chart used to elaborate the story, or an experience report that is related to the content. Having these artefacts printed out, and attached to the index card may address this need; however, such means is just not very flexible.
Fifth, if we are to produce a progress report of the states quo of the iteration progress that requires congregating a subset of attributes of each story into one document, using index cards would imply someone has to traverse through all the cards and extract as well as duplicate data from each one of them. This demonstrates that the cost of extracting and reformatting the information held on paper is higher than if it would be captured in digital formats.

2.3.4 Existing Computer-based Planning Systems

As this research focuses on the planning aspect of agile project management, I will use agile project management system and agile planning system interchangeably. Existing agile planning systems can be classified into the following categories – form-based, Wiki-based, form- and Wiki-based combined, and board-based planning systems.

2.3.4.1 Form-based planning systems

Form-based systems feature an interface that has collapsible forms, assorted buttons, check boxes, etc. The kind of interface can also be found in most standalone applications and on many Web sites. These systems usually present, collect, and store planning information in structured formats. Thus, the captured data may be used directly to derive other information. For instance, two attributes of a story that are usually collected are estimated task completion effort in hours, and actual spent effort in hours. The systems can calculate time estimation for the uncompleted portion of a story using these two attributes. Common functionalities provided by form-based interfaces include searching and sorting by specified attributes, and easy navigation between abstract and detailed
information, etc. Most agile planning systems belong to this category; for example, VersionOne [VersionOne 2005], Rally [RallySoft 2004], ScrumWorks [Danube Technologies 2000], and XPlanner [Open Source Project 2004].

![VersionOne: an example of form-based system interface](image)

#### Figure 2.2: VersionOne: an example of form-based system interface

2.3.4.2 Wiki-based planning systems

“Wiki is a piece of server software that allows users to freely create and edit Web page content using standard Web browsers. Wiki supports hyperlinks and has a simple text syntax for creating new pages and cross-links between internal pages on the fly” [Leuf and Cunningham 2002]. Newer Wiki systems come with editors that are based on visual components such as, tables, forms, fonts, etc, to render WYSIWYG (What You See Is What You Get) effect [WYSIWYG Wiki 2005] for easy content editing. A Wiki allows the organization of information to be edited in addition to the content itself. Many Wiki systems also provide features such as “classification of pages, subscription and
notification of updates, and version control” [Chau 2005]. WIKIPEDIA [WIKIPEDIA 2005], for instance, is a widely known Wiki site. The following figures demonstrate how Wiki texts can be translated into Wiki pages in JSP (Java Server Page) Wiki. For example, “[a-text | b-link]” represents displaying a-text on the screen while make it linkable to the URL b-link specifies (see Figure 2.3.a and Figure 2.4.a), and “!” means rendering the ensuing texts in bold (see Figure 2.3.b and Figure 2.4.b). Wiki-based systems feature knowledge capturing and sharing as they can capture unstructured information. On the other hand, this type of tool lacks capabilities to organize large volumes of information on one Web page. Extracting specific data out of unstructured information requires extra effort. In addition, some Wiki systems, for example: WIKIPEDIA, can be venerable to open access issues since data access authentication is not built-in by default.

Figure 2.3: Editing Wiki text using a textbox of standard Web browsers

created with the trial version of PDF-Creator.net
Testing Web-Portal Applications

With the current growth towards development of Java-based enterprise portal applications Web Portals, identifying practices that help to test portal applications in a more comprehensive and effective manner become important.

To this end we have the following major research goals:

- Evaluate the current process of testing web-portal applications.
- Gain valuable insights that identify the need for automated tool support in areas where portal applications cannot be tested at present.
- Evaluate the usability and usefulness of the WIT framework developed for In-container Testing of Web-Portal Applications.

WIT: A Framework for In-container Testing of Web-Portal Applications

Goals WIT supports writing and executing automated in-container test cases for web portal applications. Testing approach implemented by WIT allows debugging and detecting of:

- Deployment environment related problems
- Security: role-based testing of resource access

Figure 2.4: The rendered Wiki page

2.3.4.3 Form-based and Wiki-based Combined Planning Systems

Form-based systems feature information management with structured formats, and are adept at dealing with a large data set; however, the presumed formats lack the capability of capturing emerging ideas that do not fit into the predefined structure. In contrast, unstructured formats that Wiki systems employ can accommodate emerging ideas, but they are less suitable for display and organization for large volumes of information.

Benefits provided by combining Form- and Wiki-based systems are now integrated in some agile planning systems such as, MASE [E-Business Engineering Group 2005].

Figure 2.5 shows a mixture of structured (see Figure 2.5.a), and unstructured information (see Figure 2.5.b) rendered by MASE system.
Agile Project Planning and Knowledge Sharing With Horizontal Displays and Tablet PCs

Figure 2.5: MASE allows structured (a) and unstructured (b) information to co-exist on one page.

2.3.4.4 Board-based Planning Systems

Glue Wiki [Floranta] (see Figure 2.6) is a board-based planning system. Stories created are represented as image cards, and the large rectangle areas are either the Product Backlog or iterations. The system is unique in that I have not come across any other agile planning systems structuring planning information similarly. Stories can be created by clicking on an empty area within a board, and then a menu appears with card creation options. Arranging stories is easy; using click-and-drag. Two major defects prevent its use as a real tabletop interface. First, the planning board cannot be scaled to the full screen. The text around the boards constantly reminds users that this is a Web interface.

In addition, stories can only be displayed in one orientation which poses viewing difficulties for people sitting or standing at wrong sides of table.
2.3.5 Computer Media Planning – Advantages

Existing agile planning systems possess the following advantages.

First, they are typically Web-based systems, which allow stories to be accessed and maintained anytime anywhere. This is useful given the growing trend of managing software projects in a distributed fashion. Also, identity authentication can be enforced in a computer system to ensure a reasonable level of data security.

Second, stories are stored in a centralized location; which makes them easier to maintain and keep track of in comparison to paper index cards in distributed settings.

Figure 2.6: Glue Wiki represents iteration as a planning board, and story cards can be placed on the board as a metaphor that they belong to such iteration.
Third, various management metrics can be extracted from story records. For example, the number of completed stories for a past iteration or an average over- or under-estimation for stories on individual or team basis. Such metrics can be important references when it comes to project estimation and release scheduling.

Lastly, as more and more computing power is employed in modern workplaces, digital, in comparison to handwritten, information can be easier integrated into a greater variety of tasks required by execution and controlling processes of agile project management – a criterion for planning outcome storage medium that is outlined in Chapter 2.1.

2.3.6 Computer Media Planning – Disadvantages

Two factors are responsible for difficulties using existing computer-based agile planning systems during iteration planning.

First, the classic computer interaction techniques, keyboard and mouse, require more manipulation time and a higher degree of attention [Subrahmonia and Zimmerman 2000]. The paper [Subrahmonia and Zimmerman 2000] described a project aimed to alleviate the workloads of air traffic controllers by replacing manual operations with computer systems. Issues concerning the replacement include the following:

“Before selecting an item in a menu, a controller has to click in a zone which is sometimes very small. This increases manipulation times according to Fitt’s law. (As a result), the decrease in performance (editing flight strip) is noticeable.” (manipulation time)
“Manipulating menus is relatively slow, and users have to pay attention to what they are doing.... (It is in contrast) to manual manipulations (editing flight strip) that are often performed in parallel with other tasks, without paying much attention.” (degree of attention)

Training may reduce but cannot eliminate these two identified issues [Subrahmonia and Zimmerman 2000]. Agile planning encourages high bandwidth face-to-face communication to accomplish idea creation, elaboration and other coordination activities. A computer-based iteration planning tool with classic human-computer interaction does impose extra burdens to users during planning by requiring long manipulation time, and interrupt face-to-face communication by demanding high degree of attention. This is also supported by my own experiences.

Second, modern computers are designed for use by single users [Scott et al. 2003b]. Hereinafter, I refer to modern computers as “single-user computer”. Concurrent user interaction with the computers is inherently difficult. When single-user computers are used for collaborative work, the users resort to a turn-taking system access – pass of control of the input device. It is suggested that such system access deteriorates the quality of collaboration [Scott et al. 2003b] as following:

“In the paper-based sessions, all children physically manipulated pieces, and the sharing of the pieces occurred naturally. In contrast, children were less physically engaged in the computer-based sessions. Children often sat still, directing their view primarily towards the computer screen. This lack of physical engagement may impact the overall effectiveness of the collaboration, through decreased user performance, motivation and naturalness of interactions.”
Existing agile planning systems presume single-user computers as their running environments. Yet, the use of a single-user interface for collaborative work is a compromise and prior work has shown such a compromise to be inadequate [Stewart et al. 1999, Scott et al. 2003b], and can potentially deteriorate the overall effectiveness of the collaboration.

2.3.7 Combined Media Planning – Advantages

The advantages of using physical and computer-based planning media are complementary. During planning, the use of physical planning media requires little attention, and helps capture flashing thoughts and facilitates communication. After planning, stories created and edited content are then entered into a computer system, which allow stories to be persisted permanently and remotely maintained. Combined planning media incorporate the use of both physical and computer-based planning media to offer the aggregated advantages.

2.3.8 Combined Media Planning – Disadvantages

Story data needs to be converted from paper into computer systems. Data conversion is a slow and tedious job. It is usually considered a drudgery that most developers would not like to undertake. Prior research [Luff et al. 1992] revealed another downside concerning handwritten information in a computerized environment; that is, a considerable amount of duplicate information can exist on both paper and in computers; which results in a waste of time, effort, and resources.
To conclude, no existing planning tools support agile planning in full. As such there exists a need to explore new tool prototypes. The new tool should provide a collaborative interface similar to physical planning media, and be able to accommodate post-planning development activities like existing agile project management systems. The concept of Ubiquitous Computing (UC) may be able to bridge the gap between the physical and computer worlds to enable natural human interactions with embedded computer devices, and retain benefits of using computer systems.

2.4 Ubiquitous Computing (UC)

A preference for use of physical (traditional) media is a recurring theme in many collaborative tasks such as planning, scheduling, brainstorming, design and layout activities [Wellner 1993, Scott et al. 2004]. Wellner [Wellner 1993] asserted that documents live in two separate worlds: the electronic world of the workstation and the physical world of the desk. Interaction techniques with these two worlds differ in styles and functions. Choosing to interact with one world often means forgoing the advantages of the other. The following quoted paragraph from [Wellner 1993] best delineates the rationale why traditional media are preferred.

“The ways we physically interact with electronic documents are limited compared to our interactions with paper, pencils, erasers, rulers, and other traditional tools on the desk. When interacting with objects in the physical world, we take advantages of natural skills developed over our lifetimes. We use our fingers, arms, 3D vision, ears and kinaesthetic memory to manipulate multiple objects simultaneously, and we hardly think about how we do this because the skills are embedded so deeply in our minds and bodies.
Although electronic documents provide valuable functions such as spell checking, numerical calculations, and keyword searching, people must sacrifice highly developed tactile skills with paper in order to access these functions.”

[Wellner 1993] promotes tool invisibility. Here the term invisibility indicates that a tool does not intrude on users’ consciousness so that they can focus on tasks. Making use of traditional computers usually isolates the users from the overall collaborative situation. This is because too often the traditional computers remain the center of attention throughout collaboration [Wellner 1993].

“The idea of UC first arose from contemplating the place of today’s computer in actual activities of everyday life” [Weiser 1994]. The goal is to enhance “computer use by making computers available throughout the physical environment, but making them effectively invisible to users” [Weiser 1994]. The general approach to the definition and construction of UC systems is by considering everyday objects found in offices and homes, especially those whose purpose is to capture or convey information [Weiser 1994]. “The most ubiquitous current informational technology embodied in artefacts is the use of written symbols, primarily words” [Weiser 1994]. They convey information on various sizes and shapes of written surfaces. Therefore, computer embodiments may come with various sizes and shapes.

Here I illustrate the benefits of UC with an example project: i-Land. The i-Land project [Streitz et al. 1999] designs and implements innovative workspaces that are based on integration of virtual information and real architectural spaces. The project designed the following hardware devices: the Dynawall, the CommChair, and the InteraclTable, along with a software and wireless infrastructure that facilitates integration and
dissemination of information in such workspaces. The Dynawall is a touch-sensitive interactive electronic wall that can be used to create and organize information. The CommChair is a mobile chair with a built-in slate computer. The InteracTable is a mobile interactive table that is designed for creating, displaying, discussing and annotating information. The software interfaces enable natural human interactions by allowing inputs through pens and fingertips in addition to conventional input devices – keyboards and mice. Also, the software implements different gesture recognition algorithms for different devices to fulfill their interaction practices. The following are scenarios that i-Land can mediate whereas traditional computerized workspaces cannot. First, impromptu discussions in the hallway sometimes lead to explanations that may demand sketched drawing. The Dynawall can turn a seemingly traditional wall to accommodate such emerging needs right on the spot. Second, sub- and re-grouping often takes place in group work. i-Land is able to facilitate such events by merging and breaking down working artefacts automatically as the events take place.

As demonstrated, working in an environment that is empowered by the concept of UC allows people to place their primary concerns on tasks at hand rather than being restricted by how the tools are expected to be used.

2.5 Incorporating UC into Agile Planning

Traditional agile planning is typically done around a table; therefore, my research focuses on digital tabletop display and pen computing technologies. The digital tabletop is used for information organization, and the pen computing devices are used for intuitive information creation.
2.5.1 Tabletop Display Technologies

A tabletop display is simply a table that is augmented so that a part of its surface can be used as an input and/or output device [Kruger 2004]. The device is constructed so that its use can be integrated with human collaborative activities. [Scott et al. 2003a] revealed four general classes of digital tabletop systems: digital desks, workbenches, drafting tables, and collaboration tables. In their own words:

“Digital desks are designed to replace an individual’s traditional desk by integrating activities involving paper-based and digital media. Workbenches allow users to interact with digital media in a semi-immersive, virtual reality environment projected above a table’s surface. Drafting tables are designed to replace a typical drafter’s or artist’s table. They have an angled surface and usually used individually. Collaboration tables are digital tabletops that support small-group collaborative activities, such as group design, story sharing, and planning.” Agile planning requires a collaboration table. I have come across four digital collaboration tables [Streitz et al. 1999, Tandler et al. 2001, Shen et al. 2002, Shen et al. 2003] in my literature survey. Here I present the InteractTable and the UbiTable since their applications are more close to group planning.

The InteractTable [Streitz et al. 1999] is 1.15 m high, and provides a horizontal touch-sensitive display of 65 cm x 85 cm. People can write, draw, and manipulate information objects on the display with a pen and fingers along with a wireless keyboard. The table supports single input. The interaction area supports no predefined orientation. New forms of human-computer interaction – rotating and shuffling individual and groups of information objects – are implemented to accommodate easy viewing from all
perspectives. The InteractTable is designed for creation, discussion and annotation of information objects by a group of two to six people standing around it. In addition to a pen and fingers the table can also take voice-based annotations. The table supports brainstorming activities and is associated with no particular work processes.

The UbiTable [Shen et al. 2003] is designed to be a simple walk-up utility for ad hoc, impromptu face-to-face interaction. When connecting with user’s laptops, the table allows both users to fluidly share, manipulate and exchange, while maintaining explicit control over their documents [Shen et al. 2003]. The UbiTable supports public, private and personal areas [Shen et al. 2003]. Private data existing only on the users’ laptops thus is not accessible or visible to the other person. Personal data is visible but not electronically accessible by others on the table. Public data allows shared visibility and access. The following is a usage scenario of the UbiTable [Shen et al. 2003]: John and Mary meet in an coffee shop and wish to discuss Mary’s notes. They first connect their laptops to an UbiTable nearby. Once connected, the system will provide both laptops a user interface that is divided into upper and lower pane for private and personal space respectively. Mary wishes to make her notes visible to John; she places the notes from her personal space to the lower pane on her laptop, first. The notes then are available at the Mary’s side of the table but remain personal properties – that is John can see the notes but cannot access them. If Mary wants John to be able to edit the content of the notes, she would need to push them to the center circle of the UbiTable where the public area is located.
2.5.2 Pen Computing (PenC) Technologies

As computer technology evolves, different means of user input have been invented: keyboards, mice, trackballs, light pens, etc. However, none of these really fits the traditional human way of communicating thoughts and ideas [Meyer 1995]. The idea of PenC is simple: computers that are able to sense our drawings and interpret our handwritings would be much easier to use, and the ease of information manipulation would allow users to place greater concentration on the contents of their writing, shielding the users from technical restrictions [Meyer 1995]. A pen-based computer should at least consist of a flat display which records and displays the traces from a user’s moving pen [Meyer 1995]. Optional yet important to provide is the capability of recognizing human handwritings, i.e. converting scribbles to digital text [Meyer 1995].

A pen computing device should at least consist of a display, a digitizer, and optionally a handwriting recognizer [Meyer 1995]. A display is responsible for showing the traces of pen movements. A digitizer is a hardware device that is either mounted on top or under the display to capture written and drawn inputs. The tracking capability is achieved by using electronic materials, resistive films, or combinations of both. The trail of the pen is shown as a path on the display known as digital ink. The ink can be left as is, or converted to text or graphics, or represent a gesture as user’s commands. A handwriting recognizer is a software application which interprets the handwritings in ink and produces the equivalent information in the text format. Some researchers have aimed for the ultimate goal of this technology to deliver 100% accuracy in translating handwritings. However, according to Meyer [Meyer 1995], such aim is a futile attempt since even human cannot fulfill such requirements, let alone computers that are “illiterate” to human
writings. Another optional yet significant feature for Pen Computing devices is mobility. The use of pens instead of keyboards allows pen computing devices to be much smaller and portable. Augmented with wireless communications their use becomes more pragmatic.

2.6 Research Devices

An ideal planning device for this research would be a digital table that is augmented with pen computing capabilities as that is how a physical table is usually used in agile planning. Most digital tables can take pen inputs; however these tables do not support the concept of digital ink. Therefore, the handwritten inputs cannot be restored or recognized. In addition, table input resolutions are in general too low to take the kind of handwritten inputs there are usually entered on paper index cards. As such, two types of device are used for this research: the iLab table and Tablet PCs.

2.6.1 iLab Table

Figure 2.7: the iLab table

The only digital tabletop that is available for this research is custom built in the Interactions Lab at the University of Calgary [Habelski 2004]. I call it the iLab table. The table is composed of an input device – DViT [Technologies 2004] – as well as an output device – two projector and mirror sets that
are mounted on the ceiling. Both of the projects are connected to the same computer. The output device projects application image onto the surface of DViT. The surface equates to a workspace of 2048x1024 pixels and covers an area of 145.5 cm x 97 cm. The DViT screen simulates touch sensitivities by using rows of infra-red LEDs along each edges with cameras embedded in each corner to detect touched points on the surface. It can take up to two simultaneous user inputs or touches. The screen area of DViT can accommodate 6-8 adults sitting or standing around it with comfort.

Compared with other tabletop displays, the iLab table is more suitable for agile planning. For example, the ConnectTable [Tandler et al. 2001] only allows a two-people collaboration which does not match with the size of typical agile teams. The InteracTable is smaller than the iLab table and provides a lower input resolution. Making use of it for agile planning would have resulted in more significant table-size vs. team-size and less eligible tabletop writing problems. The same disadvantage also eliminates the use of Personal Digital Historian System [Shen et al. 2002] and UbiTable [Shen et al. 2003] in agile planning.

A few problems arise from using DViT as the input device [Habelski 2004]. People wearing long-sleeved or loose-fitting shirts often experience difficulties in interacting with the tabletop since their shirts and sleeves often touch parts of the surface by accident. This issue often occurs when people stand up and reach over to access an item across the tabletop. This inconvenience takes a while for the users to get used to it. Another issue is when participants grab the edge of DViT, they obscure the infra-red LEDs along the edge and interfere with the input detection. The participant needs to keep all parts of clothing away from the table edge or the surface in order to acquire a satisfactory interaction.
experience. These problems need to be resolved by proper user-training, improving input mechanism or simply letting the participants learn from their own experiences. Some other problems are related to the top projection onto the glass surface of DViT. The projecting mechanism results in minor display mistakes because some light rays will be reflected directly on top of the surface while others enter the surface and generate total reflectance [Habelski 2004]. People seem to adjust to this visual mismatch quickly. Lastly, it is hard to handwrite information with DViT input and produce a sensible digital representation of the writing. This is because writing has to be done without ones’ wrists touching the surface due to a previously identified issue. Also, the screen resolution is rather low for natural human writing despite the fact that it is one of higher input-resolution tables.

2.6.2 Tablet PCs

A Tablet PC is a laptop that includes a display and a digitizer screen. The keyboard is either detachable or hidden under the display when it is used as a notepad. A Tablet PC can take direct accurate inputs from the screen, and is equipped with a handwriting recognizer. The digitizer works with digital ink, which allows the direct manipulation of human handwriting similar to using pen and paper. Tablet PCs are mobile systems that often come with a wireless connection capability. Furthermore, they possess an ergonomic form factor – lightness and thinness, facilitating users to concentrate on human interactions even in motion. Underlying a Tablet PC is a full-fledged computer which allows the device to be used in daily work. Another feature that Tablet PCs provide is gesture recognition. *Gestures are hand movements that are used for signalling
through the air” [Meyer 1995]. Similarly, gestures in writing are hand movements that signify various intensions [Meyer 1995]. For example, striking through a word to delete a word or placing a triangular sign above the text to add a missing word. Gestures are widely accepted, intuitive to use and easy to understand [Meyer 1995]. They can be created and customized based on applications.

2.7 Natural Human-table Interaction Algorithms

Two natural human-table interactions have been explored and implemented by the Interactions Lab at the University of Calgary. These algorithms can be integrated with a tabletop-based agile planning environment to make electronic planning a lot closer to index-cards-based planning.

2.7.1 Rotation and Translation (RNT)

Through an observational study [Kruger et al. 2004] revealed that manipulation and reorientation items are commonly exercised in tabletop collaboration. This study revealed three major roles that orientation plays during collaboration:

- **Comprehension.** It is easier to comprehend objects when they are facing a viewer “straight up”.
- **Coordination.** Orientation helps to establish group and personal territories and to signal ownership of objects.
- **Communication.** Orientation can be used to initiate communication and its continuous use can be interpreted by other members about collaborative work patterns.
[Kruger et al. 2004] pointed out that traditional software considered these two actions distinct; i.e. traditionally rotation and translation were done in two separate commands. This separation can be an adversary effect according to two prior research works. [Fox et al. 2000] demonstrated that rotating and translating (moving) table items are inseparable actions in the physical word, and [Jacob et al. 1994] suggested separated interaction techniques are considered improper for integrated actions. Thus, [Kruger et al. 2005] developed an integrated interaction technique called Rotate’N Translate (RNT). RNT simultaneously rotates and translates a digital object by a single controlled point. The implementation is technology-independent, and requires no specialized hardware. Figure 2.8 demonstrates how RNT objects respond to a control point.
Figure 2.8: Illustration of RNT motion – image extracted from [Kruger et al. 2004]
2.7.2 Casual Piling

“Casual piling of paper-based media during traditional tabletop collaboration is an important practice that helps coordinate task and group interactions” [Scott et al. 2005]. The observational study [Scott et al. 2004] revealed that people use the practice of casual piling to facilitate organization and sharing of task resources. This practice involves creating piles, moving temporary and irrelevant piles to the peripheral working area and leaving the focused ones in the center, and constantly relocating piles for easier access.

“The ability of relocating piles on the table played a key role in coordinating task and group interactions during the collaborative activities” [Scott et al. 2005]. When a pile of items are moved close to a single person or a group of person(s), they reserve the temporary ownership of the items. In contrast, when a pile of items is moved to the center of a table, it usually signifies that the pile is open for public access. Individuals use piling to remind themselves or others of work not yet reviewed or completed. Casual piling is a cognitive light-weight mechanism for storing items that are otherwise difficult to classify [Malone 1983]. Such light-weight organizational methods allow people to concentrate on their tasks and interaction with other group members, since they do not have to spend time and effort on repeatedly filing and re-filing tasks [Scott et al. 2005].

Mobile Storage Bins supports casual piling. This algorithm was designed and implemented by Scott et al. [Scott et al. 2005]. A mobile storage bin features the following:

- **Container capabilities.** Storage bins provide the capability to the container allowing items to be added and removed collectively or individually from them.
• **Mobility.** Piles are often moved when someone wants easy access to their contents, or they are in the way of the main tasks. “Extending the container metaphor, all items in a storage bin could be readily relocated simply by moving the storage bins” [Scott et al. 2005].

• **Visual characteristics.** Storage bins are designed to have loose curve and adjustable boundaries. Thus, bins can be differentiated by shape.

• **Storage.** Prior work [Robertson et al. 2004, Guimbretière et al. 2001] has shown reducing an object’s size in storage area is useful. As a result, it is designed that the size of items is reduced as they enter a storage bin.

### 2.8 Single Display Groupware (SDG)

SDG is a model for supporting “collaborative work between people that are physically close to each other” [Stewart et al. 1999]. It is defined as computer programs that enable collocated collaboration via a shared computer with a shared display and simultaneous use of multiple input devices [Stewart et al. 1999]. SDG applications are different from traditional groupware systems as traditional groupware systems are intended to be run on multiple workstations and communicate with one another via a computer network, yet still relay on single input and output channels for each user [Stewart et al. 1999]. In contrast, SDG applications provide each user a private input channel through the use of separate input devices, but all users must share one output channel. Potential benefits of SDG applications are revealed as following:

• **Enabling collaboration that was previously inhibited by social barriers.** For example, in many cultures people are reluctant to invade other’s personal space.
Therefore, they refrain themselves from participating in collaborative activities when another person is in control of the input device.

- **Enriching existing collaboration on a computer.** For example, there is no need for turn-taking interaction.

- **Encouraging peer-learning and peer-teaching.** Since SDG provides multiple communication channels, it is possible to enrich learning by diminishing competition for access to the input channels.

- **Strengthening communication skills.** Strong-willed users can no longer monopolize a system by controlling the single input device. Thus, users have to communicate to resolve conflicts.

  Some potential negative effects are listed as following:

- **New conflicts and frustration may arise between users when they attempt simultaneous incompatible actions.** Working in parallel can be a disadvantage if users each have hidden conflicting agenda. Such a situation can be very problematic for content navigation.

- **SDG applications squeeze much functionality into a very limited screen space, which may result in reduced functionality.** The view of SDG applications can get very confusing at times as many are working at the same time and potentially on different objects.

- **Due to increased processing requirement, SDG applications may be slower than single user and traditional groupware systems.** It is an inevitable price to pay for dealing with concurrent user interactions.
• **SDG implementation is highly environment dependent which raises portability issues.** Successful SDG implementation depends on low-level operating system support, thus applications might not be potable.

• **Completing tasks may take longer time since there is no strong willed person who is in charge of system interaction.** It is because SDG applications give equal say to all users, no one can monopolize the input devices.

• **Users may actually collaborate less.** Since the users can work in parallel, it is possible for them to complete own tasks without communicating with others.

Prior work [Scott et al. 2003b] investigated the SDG model in practice by introducing a SGD application into a children’s learning environment. The work revealed the following findings. Forcing children to share one input device contributes to off-task behaviour and boredom. In addition, children appreciate, enjoy and take advantages of concurrent input support whenever possible when it is provided. These findings seem to indicate that collaborative technology may enhance social interaction and provide benefits.

### 2.9 Summary

Outcomes that are generated by software project planning are usually used to support various activities as illustrated in the PMBOK. Plans are important to agile methods since they anticipate and encourage changes, and the plans are used to integrate and coordinate project activities. Currently there are two types of planning media and three derived tool usages. In this chapter I went though each of them and discussed their pros and cons. It was concluded that no existing tools support agile planning in full. Drawing from the drawbacks of each planning approach regarding tool usage, I can conclude an ideal tool
support for agile planning should provide the benefits of physical planning media to
mediate face-to-face collaboration. As well the tool should be an integrated resource to
agile planning so that a plan can be made available via the Internet to support distributed
access. In addition, the cost of translating handwritten inputs into plain texts should not
be induced on developers as data entry does not fit with their primary responsibilities.

UC is a concept of embedding computer support into the background of a work
environment. It enables natural human interactions, and uses them as inputs so that users
can concentrate on tasks rather than tools. Combining digital tabletop and pen computing
technologies may result in a tool system that renders the benefits of both physical and
electronic planning media. This research adopts the iLab table and Tablet PCs to
construct a collaborative environment for agile planning. In terms of my research
application, SDG is considered to be a reference implementation model that aims to
facilitate collaborative work. The setup is similar to using the iLab table as a common
output channel with multiple Tablet PCs separate input channels in an agile planning
environment. Prior work [Scott et al. 2003b] revealed the potential social and
collaborative benefits for such setup. In the next chapter I will identify requirements
essential for digital tabletop based tools to facilitate agile planning.
CHAPTER 3. TOOL CRITERIA & ASSESSMENT

As presented in the previous chapter, existing tools provide limited support to agile planning. An ideal tool should have a physical planning media-oriented user interface while retaining the functionalities of a computer system and eliminating costs of data conversion. Thus, I propose an agile planning system that is based on the concept of UC.

In this chapter, I first outline specific requirements for such a planning system. Then a critical evaluation of existing agile planning tools is presented in terms of their abilities to support the identified requirements. The result of this evaluation provides the motivation to build the proposed proof-of-concept tool, AgilePlanner - details of which are presented in Chapter 4.

3.1 Tool Requirements for Agile Planning

As explained in Chapter 2.5, it is my intention to realize a natural agile planning setup that is based on index-cards and tabletops. In the following sections, I outline a list of tool requirements for such a planning system. These requirements are divided into two categories – functional and interaction-related. The functional requirements focus on enabling functionality to complete activities prescribed by agile planning processes such as creating and editing index cards, etc. The interaction-related requirements focus on supporting interpersonal interaction by enabling intuitive human-system interaction for organising and entering agile planning information. The agile planning system should provide the following functional capabilities – requirement 1 and 3 are derived from
perusing agile methods literature [Abrahansson et al. 2002, Larman 2004] and the rest of requirements are produced through my own experience:

1. **Agile planning objects creation and editing.** Lying at the core of agile planning are the activities of creating and editing stories, and creating iterations. Without this functionality, the system cannot complete the agile planning process.

2. **Visual characteristics for different types of story.** There are three types of stories in agile planning – bug fixes, enhancements, and features. According to my conversations with other agile practitioners, it is considered beneficial to have distinct visual patterns for them to distinguish the type of a story on the tabletop.

3. **Agile planning metrics management.** Past iteration experiences make credible reference points of future development. Metrics preserve different aspects of past iterations; for example, metrics that would reveal overall team performance over previous iterations. Other metrics may include, time estimations of a task in the best, most likely, and worst case scenarios.

4. **Planning for multiple iterations.** Allowing planning for multiple iterations at the same time allows for a deeper look into the future of an agile project. This way it is easier to coordinate stories between the iterations during planning.

5. **Systematic organization of planning objects.** A display provides only limited screen area that can be easily swamped with a large number of stories from an agile plan; therefore, systematic story organization can be beneficial.

6. **Fluid transition between plan changes and the consequent results.** Instant and effective visual replies for changes made to an agile plan can streamline planning collaboration.
7. **Team and identity authentication.** The agile planning system should enforce an authentication protocol so that it is capable of keeping iteration plans secure at all times.

8. **Real-time exposure of the plan via the Internet.** This feature enables remote access and maintenance to the plan, and supports distributed software teams. In addition to the functional capabilities mentioned above, the agile planning system should also support interaction-related capabilities. Interpersonal interaction primarily relies on engaging conversation. When it is interfered by tool operations, collaboration breakdowns can arise [Elwart-Keys et al. 1990]. Thus, the following list of requirements focus on reducing tool interferences to planning. These requirements are based on [Scott et al. 2003a]’s work that provides guidance for designing tabletop systems to support collaboration.

9. **Simultaneous planning information organization.** Agile planners may simultaneously organize stories on a table surface. Existing planning systems do not support multiple inputs. As a result, the planners are forced to take turns and check the availability of the systems before accessing them. This introduces cognitive overheads upon the users [Scott et al. 2003b]. Supporting concurrent interaction style is more suitable for organizing collaborative story information on a tabletop.

10. **Simultaneous planning information editing.** Here the term planning information editing refers to activities of creating and editing story content. Agile planners may also exercise such activities simultaneously during planning. Based
on the same rational behind the previous requirement, the simultaneous user
interaction style should be adopted for the agile planning system.

11. **Story editing with handwriting inputs.** The planning system that allows for
handwriting inputs enables users to place greater concentration on creating and
ing the content of the plan.

12. **Handwriting recognition capability.** In addition, the planning system should
allow handwriting inputs. A recognition capability could translate such inputs into
ASCII texts. This can provide potential benefits as process information encoded
in text is easier than in image.

13. **Fluid transition between individual and collaborative work in agile planning.**
Story editing activities are usually considered individual work in agile planning,
while information organization activities are considered collaborative work.

Based on the results from previous research [Elwart-Keys et al. 1990,
Mandviwalla and Olfman 1994], [Scott et al. 2003a] suggested fluid tool
transitions between individual and group work could better mediate team
collaboration. Thus, the agile planning system should facilitate such transitions.

14. **Fluid transition between agile tabletop collaboration and external work.** “Co-
located group interaction is only one part of collaborative activity, thus group
members must be able to incorporate work generated externally to the tabletop
system into the current tabletop activity” [Scott et al. 2003a]. Thus, the agile
planning system should provide easy transitions between core planning activities,
and digital work that is prepared externally.
15. **Flexible user arrangement.** During a tabletop collaboration, people sit around a table at a variety of locations, both in relation to the table and other group members [Scott et al. 2003a]. Physical properties of the table, such as its size and shape, can influence user arrangements. Flexibility in user arrangements allows people to keep their personal preferences of staying at least at an “arm’s length” from each other [Hall 1966]. In addition, adults usually prefer face-to-face or corner seating positioning for having conversations [Sommer 1969]. Therefore, the agile planning system should weight the physical properties of the digital tabletop in relation to typical sizes of an agile team.

16. **Shared access to digital objects in agile planning.** The system can decide to model the following planning artefacts such as stories, iterations, and the Product Backlog, as digital objects. Based on the results from prior research [Suzuki and Kato 1995], [Scott et al. 2003a] pointed out “the benefits of using shared object can help maintain the group focus and facilitate team awareness since body positioning and eye gaze of the group members attending to the same object can be easily and clearly interpreted by members who try to catch up with the group collaboration”. Otherwise, “a gesture made to one copy of the object forces the other group members to perform a spatial translation,” [Scott et al. 2003a] introducing cognitive overheads [Gutwin and Greenberg 2000, Tang 1991].

17. **Use of physical planning objects.** Physical objects on a table can be categorized into task-related and non-task-related objects [Scott et al. 2003a]. In agile planning, task-related objects are paper index cards, and non-task-related objects can be notepads, laptops, coffee cups, beverages, etc. In general, a seamless
integration between task-related physical objects with virtual collaborative
tasks allows humans to apply their years of experience of physical object
manipulation [Scott et al. 2003a]. Non-task-related objects should be recognized
but not interpreted by the system. For example, when one puts a coffee cup on the
surface of the tabletop the planning system should not display relevant
information in the space occupied by the cup [Scott et al. 2003a].

In total, I identify seventeen requirements for an agile planning system that aims to
facilitate planning collaboration. The first eight requirements focus on assisting core
planning activities, and the last nine requirements attempt to integrate the use of such a
system during the planning process. Agile planning systems that fulfill these
requirements could conceptually be able to allow users to alternate with ease in their
interactions between other users and the integrated computer system.

3.2 Existent Tool Support Assessment

As presented earlier, there are two types of planning media for agile planning: physical
planning media and computer-based planning systems. In the following sections I present
an evaluation of the existing planning tools using the tool requirements identified above.
The identified criteria are essential for tool support to comprehensively enable agile
planning. The purpose of these sections is to illustrate the current state-of-the-art for agile
planning tools. Existent agile planning systems that are included in this evaluation are
XPlanner, VersionOne, ScrumWorks, RallySoft, Wiki, MASE, and Glue Wiki. A more
detailed explanation about all of these planning systems is available at the Chapter 2.3.4.
A summary of this assessment is presented in the Table 3.1. In this table, a (√) mark
indicates that a tool fulfills a specific requirement whereas a ( X ) is given instead when the evaluation turns out to be negative. If a tool appears to provide elementary but insufficient support for a specific requirement, it is marked with a ( √ / X ).
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Criteria Sub Categories</th>
<th>Physical Planning Media</th>
<th>Xp Planner</th>
<th>VersionOne</th>
<th>Roll/Suit</th>
<th>Wiki</th>
<th>MASE</th>
<th>Guo/Med</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile planning objects creation and editing</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Visual characteristics for different types of story</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Agile planning metrics management</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Planning for multiple iterations</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Systematic organization of planning objects</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fluid transition between plan changes and the consequent results</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Concept of team and identity authentication</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Real-time exposure of the plan via the Internet</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Functional Requirements**

| Simultaneous planning information organization                          |                         | ✓                        | ✓           | ✓          | ✓         | ✓    | ✓    | ✓      |
| Simultaneous planning information editing                               |                         | ✓                        | ✓           | ✓          | ✓         | ✓    | ✓    | ✓      |
| Story editing with handwriting inputs                                   |                         | ✓                        | ✓           | ✓          | ✓         | ✓    | ✓    | ✓      |
| Handwriting recognition capability                                      |                         | ✓                        | ✓           | ✓          | ✓         | ✓    | ✓    | ✓      |
| Fluid transition between individual and collaborative work in agile planning |                         | ✓                        | ✓           | ✓          | ✓         | ✓    | ✓    | ✓      |
| Fluid transition between agile tabletop collaboration and external work |                         | ✓                        | ✓           | ✓          | ✓         | ✓    | ✓    | ✓      |
| Flexible user arrangement                                               |                         | ✓                        | ✓           | ✓          | ✓         | ✓    | ✓    | ✓      |
| Shared access to digital objects in agile planning                      |                         | ✓                        | ✓           | ✓          | ✓         | ✓    | ✓    | ✓      |
| Use of physical planning objects                                       |                         | ✓                        | ✓           | ✓          | ✓         | ✓    | ✓    | ✓      |

Table 3.1: Assessment for existing tool support for agile planning.
3.2.1 Tool Assessment by Functional Requirements

The following is an evaluation of existing agile planning tools against the functional requirements identified in Chapter 3.1. Physical planning media can easily satisfy the requirement of supporting story creation and editing as well as iteration creation. In terms of providing distinct visual patterns for different types of stories, different kinds of index cards can be prepared prior to a meeting to fulfill such a requirement. In terms of assisting agile planning by managing planning metrics, it is not feasible for passive physical planning media to provide such a dynamic feature. As for assisting transition between making plan changes and viewing the consequent results, physical planning media can mediate humans to fulfill this requirement because they enable humans to apply common tabletop experiences. For instance, since human attention is usually placed on the center of tabletop, changed stories are naturally placed in that area so that everyone can be aware of the event. Changes and their results can be displayed in such a way to appeal to human viewing habits; which makes the transition fluid. Lastly, planning in collocated setting, people can see who creating or editing stories are. Therefore, these changes can only be made by users with proper access. This means the use of paper index cards upholds the criteria of supporting the concept of team and identity authentication.

All the existing planning systems support agile planning object creation and editing. Regarding the capability of displaying different types of stories with distinct visual characteristics, most of the systems except Glue Wiki use small image icons to represent the type of a story. These icons can be easily and usually are omitted when a large
number of stories are listed in the window. Glu Wiki does not support this capability at all, as the system treats all stories the same way.

With respect to support for managing planning metrics, each planning system offers assistance to certain extent. A common metric that is supported by all systems is to sum collective efforts for all stories under one iteration. Glue Wiki offers a unique metric that sums up collective work estimation over a random selection of stories. VersionOne can generate an effort sheet reporting each team member’s estimated working hours in the current iteration. ScrumWorks produces a report of the average work effort put into past iterations and makes suggestions that future iterations to be the same; seemingly dismissing the fact that the length of an iteration can very. XPlanner provides a chart which compares developers’ cumulative progress with original work estimation; however it is difficult to read team performance out of the chart produced by XPlanner. The system also tracks the start and completion date of a task. MASE tracks average completion of past iterations as a percentage, and calculates past team performance based on that. It also suggests estimated work hours to be committed to an iteration based on the past team performance and the sum of ideal work hours available for all developers. In addition, MASE tracks three types of story estimates: worse-case, most-likely case, and best-case.

In terms of enabling agile planning with multiple iterations, only Glu Wiki does not satisfy the requirement. With regard to providing systematic organization of agile planning objects, all the form-based systems are not concerned with the criterion since stories are always organized in a tree structure. It is only Glu Wiki that needs such organization; however, the functionality is not supported by the system. As for supporting
fluid transition between plan changes and the consequent requests, most existing
planning systems lack flexibility to allow their interfaces to be dynamically customized to
suite the viewing habits of humans. As a result, working with these interfaces requires
adaptations. These adaptation efforts usually are reduced after some time of use.
Therefore, I assess the existing systems provide a compromised support to this criteria.
With respect to features like supporting the concept of team and identity authentication as
well as real-time exposure of agile plan via the Internet, all the existing planning systems
provide support to both of them.

3.2.2 Tool Assessment by Interaction-related Requirements

In assessing physical planning media against interaction-related requirements, the tools
obviously fulfill all of the identified requirements with one exception. After all, these
requirements are inspired by the ease of use of physical planning media in agile planning,
and designed to let computer-based tools simulate them. The exception is to support the
handwriting recognition.

The existing agile planning systems are all constructed based on “single-user
computers”, a notion that I described in the Chapter 2.3.6. The interface of such systems
does not allow concurrent user inputs. None of them attain the use of pen computing
devices; thus zero handwriting input and recognition support. None of them incorporate
the use of digital tabletops. Hence, no support to interaction-related requirements is
provided. In addition, since there are no digital tabletops and no concurrent input support,
user arrangement with the existing planning systems is always subject to the locations of
the input devices; which is the opposite to being flexible. The only interaction-related
requirement that all the existing planning systems satisfy is to provide shared access to
digital objects in agile planning. According to my own and others experience, planning
with the existing planning systems is usually augmented by some projection devices to
provide better viewing. By virtue of such setup, only one instance of the plan exists and is
shared among all team members. With respect to the capability of incorporating the use
physical planning objects (cups, beverages, etc.) to support planning, the existing agile
planning systems also do not facilitate such possibility.

To sum the evaluation of existing tool support for agile planning against the
identified tool criteria, none of all the tools investigated provide full support for all the
seventeen identified criteria.

3.3 Summary

In this chapter, I have outlined a list of requirements for tool support that aims to
empower computer-aided agile planning with the look and feel of physical planning
media. The requirement list contains seventeen criteria that are composed of two
subcategories – functional and interaction-related requirements. Based on these seventeen
criteria, I evaluated existing agile planning tools. Physical planning media fail to facilitate
distributed access to and automatic management of agile plan information. The existing
agile planning systems have also been shown inadequate because they do not facilitate
face-to-face collaboration. The state-of-the-art tool support to agile planning is clearly
proven insufficient by this evaluation. In the next chapter, I will present the design and
implementation of a novel tool that aims to (1) facilitate face-to-face collaboration during
iteration planning as in traditional tabletops, (2) provide remote planning information
maintenance as in existent agile planning systems, and (3) alleviate users from the labour of data conversion between human handwriting and plain text.
CHAPTER 4. AGILEPLANNER: PROOF-OF-CONCEPT

The previous chapter identified a list of criteria that were needed to be fulfilled by an agile planning environment based on digital tabletops. Based on these criteria, I present a proof-of-concept planning environment, AgilePlanner. First, I describe the functionality that each component in AgilePlanner provides. Next, I demonstrate the effects of the planning environment with a usage scenario in agile planning. Then, I present the technical details underlying the functioning of such an environment. The following sections use agile planning and iteration planning as interchangeable terms based on explanation provided in Chapter 1.2.

4.1 Tool Functionality for Agile Planning

![Diagram of AgilePlanner](image)

Figure 4.1: AgilePlanner – a computer-aided environment for iteration planning.

AgilePlanner is a computer-aided environment for iteration planning. The environment extends on our existing planning system, MASE. Users can perform natural collaborative
activities using the iLab table (see Figure 4.2) and individual activities using Tablet PCs (see Figure 4.3). The iLab enables natural human interaction using the table, allowing stories to be organized with fingers and pens. AgilePlanner incorporates these collaborative features of the iLab table by extending the algorithms and implementations – RNT [Kruger et al. 2004] and Mobile Storage Bins [Scott et al. 2005] – created at the Interacts Lab at the University of Calgary. Here onwards, these algorithms and implementations will be referred collectively as the iLab table software. The software suite forms the foundation of AgilePlanner. The Tablet PCs enable the users to create and edit story content with handwritten inputs, and programmatically translate them into ASCII texts. They can then be persisted by MASE. As MASE is Web accessible, the persisted information can be ubiquitously maintained and manipulated to satisfy various needs during development as discussed in Chapter 2.1.

Figure 4.2: An agile team using AgilePlanner to conduct iteration planning.
4.1.1 Remembering Previous Planning Results

AgilePlanner represents stories as images on the surface of the iLab table. The locations of all cards in relation to the surface of the table are recorded by the software. This functionality serves as a reminder of what has happened in the last planning session, because “when a person learnt the locations of objects in an environment, they typically also acquire non-spatial information, for instance, information about activities which have been performed at a particular locations” [Mecklenbräuker et al. 1998]. Moreover, this functionality makes finding the specific stories easier when humans revisit the table.

4.1.2 Simultaneous Index-card-like Story Creation and Editing

With AgilePlanner, creating and editing stories is similar to using pen and paper index cards. It is done through a story editor on a Tablet PC. The story editor can be launched by placing story creation commands available on the floating main menu, or double-clicking on stories listed under an iteration. The editor (see Figure 4.4) contains detailed information about a story and can be used to edit values associated with each attribute.
Editing can be done by either typing or handwriting. When it is done, the story updates can be released by clicking on the “Save” button (see Figure 4.4.a), at which point the handwriting inputs are translated into ASCII texts and being send to the MASE server for persistence. In addition, AgilePlanner allows multiple Tablet PCs to connect to the iLab table, thus story creation and editing can be done with any of these Tablet PCs simultaneously even when the users are in motion. If the story editor is closed by accident; a message window will be launched to prevent undesired data loss. Figure 4.4 also manifests the likeness of the look of the story editor and paper index cards in human handwriting.

![Figure 4.4: User interface of the story editor in comparison to paper index card in human handwriting](image)

4.1.3 Systematic Story Organization

In the case where a large number of stories exist in the current plan and are placed all over the tabletop, finding a particular story becomes challenging. Such challenges may interfere with human interpersonal interaction and lead to diversions from collaboration
occasionally. AgilePlanner mediates these situations by providing a systematic story organization capability that is able to reorder all the stories on the tabletop instantly upon user commands.

![Figure 4.5: The view of AgilePlanner on the tabletop before (left) and after (right) systematically reordering.](image)

4.1.4 Visual Characteristics for Different Types of Story

Stories can be of three types: bug fix, refactoring request, or new feature. To easily differentiate among these types, AgilePlanner associates distinct visual characteristics with each story type as demonstrated in Figure 4.6. The color *khaki* and the *bug* watermark are associated with bug fix stories. The color *light sky blue* and the *magnifier* watermark are associated with a refactoring request, and the color *honeydew* and the *document* watermark are associated with feature stories. These visual characteristics can be used to unravel hidden patterns. For instance, the team may discover potential incomprehensiveness of their current testing process when most of the stories under
current iteration are color coded as bugs. The same pattern may also be interpreted as bad system design.

![Image](image.png)

**Figure 4.6: Bug, refactor, and feature stories from left to right.**

### 4.1.5 Easy Navigation and Prioritization on Tablet PCs

AgilePlanner presents the same amount of information on the iLab table and on Tablet PCs in different structures. Due to the limited screen area that Tablet PCs have, the structure employed needs to be effective for information browsing and space management. In AgilePlaner, Tablet PCs are designed to be used as an individual’s workspace that allows planners to create and edit stories. AgilePlanner lists planning iterations and their children stories in a tree structure. Each iteration entry in the tree is a collapsible item providing flexibility to browse through the tree. Prioritization can be done by drag-and-dropping stories among iterations. Figure 4.7 demonstrates the tree representation that contains all necessary agile planning materials – stories, iterations, and the Product Backlog.
Figure 4.7: On Tablet PCs, AgilePlanner shows all the iteration selected for current planning and their children stories in a tree structure.

4.1.6 Fluid Transition from a Table PC to the iLab Table

On TabletPCs, AgilePlanner categorizes two listing: public and private (see Figure 4.6). The public list contains story items that exist on both MASE and the iLab Table. The private list is a temporary storage for newly created / edited stories before they are made public i.e. available to MASE or on the tabletop. Once the story creator finalizes the story content, it can be dragged into an iteration listed in the public listing window. From that point on, the created stories are available on the table at a predefined location for collaboration and on MASE for browsing. The private list is like a “playground” for users to experiment with the story content, and allows them to decide which stories should be
published and at what time. This is done to prevent premature story releases, so that public story management can be kept easier on tabletop.

4.1.7 Story Prioritization and Organization Using a Pen and Fingertip on Tabletop

AgilePlanner adopts *RNT* objects [Kruger et al. 2004] as the basis for its image story cards. These cards inherit the communicational properties of RNT objects, taking the advantages of the integrated interaction techniques for image rotation and translation and the benefits of free object orientation to tabletop collaboration. Furthermore, AgilePlanner adopts *Mobile Storage Bins* [Scott et al. 2005] as the basis to its causal piling mechanism. To make these two advanced tabletop interaction techniques useful to agile planning; AgilePlanner needs to define their interaction semantics. For example, iterations are represented as mobile storage bins. Moving an image card into a storage bin signifies adding a story to the iteration. As selected story(s) are placed in a storage bin, managerial metrics associated with the iteration are automatically updated. Currently captured metrics are the sum of all iteration stories including best case, most likely, and worse case estimations. Figure 4.8.a and Figure 4.8.b demonstrate such updates.

In addition to abovementioned capabilities, AgilePlanner also allows story card(s) to be thrown, resized and selected at once. The three interaction techniques extend on the implementation implemented by [Scott et al. 2005]. AgilePlanner augments these interaction techniques with iteration planning semantics, for instance, throwing a selection of story cards into an iteration area represents adding them into that iteration. Also, when resizing a story AgilePlanner adjusts the content of the story cards to allow it to be viewed easily by more people.
Figure 4.8: The left and right images contain the same set of story cards. The image cards in the green area are smaller than cards out of it by comparison. As stories are placed into the iteration, managerial metrics are updated demonstrated by (a) and (b). The round-shaped areas, free rotation of the image cards and the shirking effect on the cards inside iteration is produced by the iLab table software.

4.1.8 Real-time Exposure of Planning Information and Distributed Maintenance

AgilePlanner defines a communication infrastructure that is capable of synchronizing all distributed planning units: the iLab Table, Tablet PCs, and MASE. Distributed planning can thus be achieved; however, the current implementation of AgilePlanner presents many usability issues in distributed planning, and no evaluation has been conducted. In a collocated setting, AgilePlanner allows the planning participants to focus on human collaboration with natural tabletop interactions. At the same time, the software transparently records planning decisions and persist them on MASE, which makes them
available on the Internet immediately. Using regular Web browsers, distributed team members are then able to view and maintain the iteration plan.

Figure 4.9: MASE Web interface displays agile plan information.

4.1.9 Mobile Main Menu

One of the collaborative characteristics that a physical table has is once people settle at their comfortable spots they rarely move. Collaboration is easier for it because planning around a table helps them to quickly locate each other. On the iLab table, such flexibility can be facilitated by a mobile window that provides menu options. Had the manual window been fixed to an area on the tabletop display, the users would have been forced to move around in order to access the provided options; obscuring the benefits of using a
table. The menu provides functionalities such as story and iteration creation, systematic story organization, etc.

4.2 Usage Scenario

An iteration planning meeting commences by creating a new iteration. During creation, AgilePlanner prompts for constituent information for an iteration including the name, the end date, and the high-level description that describes the goal for the current iteration. The new iteration is usually placed at the center of the tabletop where everyone can easily view and access it. If preferred, the iteration storage area can be enlarged to accommodate more stories. Usually before the meeting, developers are supposed to enter their work hours spent on each story item. In cases where the developers are not able to do so, they can update the story metrics with AgilePlanner using Tablet PCs without disturbing others. These updates take effect immediately on the tabletop. The customers start extracting stories that are related to the goal of the current iteration from the Product Backlog. They usually pile the stories in the tabletop space that is right in front of them. If new stories are needed, they can be created using AgilePlanner on Tablet PCs. As sometimes the customers might go through multiple drafts creating stories in order to make them brief and focused as well as independent of other stories, multiple versions of the new story may be produced and stored under the private list using the interface as depicted in Figure 4.6. As the story creator finalizes the content, he/she can then release the story to the tabletop planning workspace by dragging the story from the private pane to the public pane using the same interface. Prioritization takes place after all the stories related to the goal of the current iteration are identified or created. The customers go
through each one of the stories according to their business significance and explain the
meaning behind each story. The developers contribute to this discussion by providing
technical assessments. Stories displayed by AgilePlanner can be naturally and easily
moved around, resized, thrown, and rotated to mediate human communication. During
collaboration, the developers can provide estimates to the stories using the Tablet PCs.
As the selected stories are moved into the current iteration, the stories shrink in size so
that the iteration area can accommodate more stories. The estimates for the iteration are
automatically summed as demonstrated in Figure 4.8. If the estimated effort to complete
the current iteration exceeds the expected iteration work hour, the team then discusses
which of the selected stories should be left out for future implementation, and remove
those image stories from the iteration accordingly. The iteration planning meeting ends
after this discussion is completed.

4.3 System Architecture

AgilePlanner consists of two major software components: AgileTable and AgileTablet.
They enable human-table and pen-and-paper interactions with the iLab table and Tablet
PCs respectively. A portion of MASE is augmented so that it can coordinate and persist
updates made from all distributed planning devices (AgileTables and AgileTablets).
Figure 4.10 illustrates the high-level architecture of the AgilePlanner environment.

Microsoft’s Windows XP Tablet PC Edition is currently the only operating system that

1 This section contains similar content available at a paper called “An Environment for Collaborative
Iteration Planning” Liu, L., Ergogmus, H. and Maurer, F. (2005), ‘An Environment for Collaborative
Maurer of a leading conference of the Agile community.
supports pen computing and ink-smart applications on PCs. Therefore, the client
component of AgilePlanner (AgileTable and AgileTablet) that implements the front-end
functionality currently runs only under Windows XP Tablet PC Edition. AgilePlanner is
entirely written in C#. AgileTable and AgileTablet share AgilePlanner Client Core that is
composed of three parallel threads as Figure 4.10 depicts. The major distinctions between
AgileTable and AgileTablet are their presentation logics and availability of support for
digital ink. The operating system provides the necessary functionality to translate ink
inputs from Tablet PCs into their textual form. Communication between the tabletop and
the Tablet PCs to MASE are realized via Web Services using the SOAP protocol. MASE
is responsible for all back-end functionality, including storing the planning data in a
relational database. MASE broadcasts update notifications through the callback of an
asynchronous service request that is placed upon the server by the client. The
AgilePlanner client is then responsible for updating their graphical interface to reflect the
latest changes. MASE treats AgileTable and AgileClient in the same way. Outside the
environment, the stored planning data is accessible to regular desktop clients through
MASE’s Web interface for asynchronous project management activities.
Figure 4.10: The architecture of AgilePlanner Client Core (left) and AgilePlanner (right), as well as a demonstration of a typical execution sequence of the system.

4.4 Implementation Details

This section elaborates the implementation details of the AgilePlanner by illustrating a typical execution sequence (numbers match those in Figure 4.10). This execution sequence starts with users creating a handwritten story.

1. When AgilePlanner is launched, immediately three parallel execution threads that are responsible for interface control (Presentation Thread), updating send-offs (Update Registration Thread), and updating reception (Update Reception Thread) become activated. Update Reception Thread first places an asynchronous service call to MASE, through which an AgilePlanner client registers itself for a callback notification. This means when a planning update is complete, MASE sends a
notification back to the client. As it is an asynchronous service, the clients can carry on their interactions with users before the callback notification returns.

2. A user launches the story editor to create a handwritten story. The Presentation Thread deals with this request by showing the story editor and getting ready to take the user’s handwriting.

3. Once the user saves the content of the created story, user handwriting is translated into plain texts, and their pictorial representations are encoded in a compressed textual representation that enables image reconstruction for the future. Both the plain text and the encoded image strings are placed into a XML [WIKIPEDIA: XML 2005] document. The XML documents are then pushed onto a queue that resides in the Update Registration Thread waiting for send-offs.

4. The Update Registration Thread checks for any entries in its queue. If there are, it traverses through the queue and places a service that corresponds to the type of update. For instance, “create new story” and “update card location” requests will require different service invocations.

5. The update message arrives at MASE, and the content is first persisted into the backend relational database. Then, MASE constructs a callback message and completes the asynchronous service requests that are previously issued by the AgilePlanner clients at the step 1.

6. The callback messages are sent back to the AgilePlanner clients. The AgilePlanner client will then first verify if the update is generated by itself. If yes, then the update is dismissed. Otherwise, the message is placed into a queue that is accessible to the Presentation Thread.
7. In the mean while the Update Reception Thread repeats the step 1, waiting for the next updates.

8. The Presentation Thread checks the queue inside of Update Reception Thread to determine if any changes have been made.

9. The Presentation Thread deciphers the callback messages and refreshes the graphical interface with the latest planning information.

10. & 11. Through the Web interface of MASE, users can access the updated planning data with standard Web browsers and the HTTP protocol.

4.5 Summary

In this chapter, I have presented the functionalities as well as the design and implementation of AgilePlanner. AgilePlanner is composed of AgileTable, AgileTablet, and the Web Services portion of MASE. The iLab table software enables natural human-tabletop interaction and effective use of tabletop space by providing RNT [Kruger et al. 2004] and Mobile Storage Bins [Scott et al. 2005] implementations. AgileTable extends the iLab tablet software with planning semantics to emulate the use of a traditional table and paper index cards with the iLab table during agile planning. AgileTablet simulates the index cards creation and editing using pen and paper with Tablet PCs. The presentation design on Tablet PCs facilitate plan navigation and story creation by utilizing a tree data structure and providing a story editor whose interface that looks like paper index cards. MASE coordinates the communication among all planning units. AgilePlanner implements an agile planning environment with iLab table and Tablet PCs.
with an attempt to enable planners to apply their index cards planning experiences in
the computerized planning environment.
CHAPTER 5. AGILEPLANNER: QUALITATIVE ANALYSIS

The objective of this research is to improve the effectiveness of tool support for agile planning. In chapter 2, I concluded that a planning tool that supports physical planning media interaction, eliminates handwriting-to-digital-text conversion effort, and provides distributed maintenance capabilities would be ideal for agile planning. As a result, seventeen criteria are elicited as essential features for such a tool. In this chapter I present the results of an evaluation of AgilePlanner against the previously identified criteria.

Table 5.1 shows the summary of such an evaluation. In the table, a (✓) mark indicates that a tool fulfills a specific requirement whereas a (X) is used to show when the evaluation yields a negative result. If a tool appears to provide elementary but insufficient support for a specific requirement, it is marked with a (√/X). Through this evaluation, it is revealed that AgilePlanner provides complete or partial support to most of the identified features. In comparison to existent agile planning tools, AgilePlanner excels because it fulfills a greater number of requirements.
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Criteria Sub Categories</th>
<th>AgilePlanner</th>
<th>Physical Planning Media</th>
<th>XP Planner</th>
<th>VersionOne</th>
<th>Rally/Gett</th>
<th>Wiki</th>
<th>MASE</th>
<th>GitWeb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile planning objects creation and editing</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Visual characteristics for different types of story</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Agile planning metrics management</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Planning for multiple iterations</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Systematic organization of planning objects</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Fluid transition between plan changes and the consequent results</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Concept of team and identity authentication</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Real-time exposure of the plan via the Internet</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Simultaneous planning information organization</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Simultaneous planning information editing</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Story editing with handwriting inputs</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Handwriting recognition capability</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Fluid transition between individual and collaborative work in agile planning</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Fluid transition between agile tabletop collaboration and external work</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Flexible user arrangement</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Shared access to digital objects in agile planning</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Use of physical planning objects</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Table 5.1: Assessment of AgilePlanner by the 17 tool support criteria in comparison with existent planning tools.
5.1 Supported Criteria

In terms of fulfilling the functional criteria for agile planning, AgilePlanner facilitates creating and editing stories by providing a story editor interface that resembles actual index cards and accepts handwritten inputs. This implementation of close-to-index-card interface and interaction allows users to create and edit stories while communicating. Although existing agile planning systems all support this requirement, their support is restricted to keyboard and mouse inputs only. In terms of iteration creation, using AgilePlanner is similar to using the existing systems.

AgilePlanner discerns among three types of stories – bug fix, enhancement, and feature – with different water marks and card colors. The distinctions are visually noticeable. AgilePlanner’s fulfillment of this requirement is better than that of paper index cards due to paper’s lack of adaptabilities to change its appearance on the fly. This fulfillment also exceeds the approach that the existing planning systems primarily take – that is to use icons to differentiate story types, because icons are generally small, and difficult to differentiate in large numbers.

In terms of support for metrics management, AgilePlaner provides the majority of the managerial metrics that MASE utilizes for tracking on its planning interface. These metrics are cumulative best case estimate, most likely estimate and worse case estimate for all the stories under one iteration. The metrics are updated automatically as iterations are changed.

As for support to multiple iterations planning, AgilePlanner allows iterations with all their stories to be moved around as a single entity on the tabletop. In addition,
AgilePlanner facilitates story transfers by allowing the iterations to be positioned adjacent to one another so that these transfers may be done in one system interaction. In comparison to the Web-form based interface of existing agile planning systems, they require two separate system commands (selecting stories for transfer and then moving them to another iteration).

As for supporting systematic organization of planning objects, Chapter 4.1.4 described this feature of AgilePlanners. In terms of mediating transitions between entering changes to plans and viewing the consequent results, AgilePlanner, like physical planning media, provides users the flexibility to specify the location of the consequent results rather. As a result, the users do not have to adapt to locations predefined by software interfaces.

With respect to enabling team concept and identity authentication, AgilePlanner coordinate with MASE to grant data access to valid team members. With respect to real-time exposure of agile plans via the Internet, this has been demonstrated and explained in Chapter 2.1.9.

For support of interaction-related criteria, AgilePlanner focuses on not interfering with face-to-face interaction. The tool enables various natural interaction techniques that physical planning media possess, including RNT, throwing, casual piling, handwriting to manage story content, etc. For instance, adding or removing a story from iterations is accomplished through the simple act of moving the digital card in or out of an area that represents the iteration. Also, creating a story can be done in a similar means to using pen and paper with AgilePlanner. As such, the use of AgilePlanner introduces little interference during face-to-face interaction.
AgilePlanner supports simultaneous story editing. This is done by utilizing multiple Tablet PCs. Prior studies [Bly 1988, Tang 1991] have revealed that people do not distinguish the following activities as being distinct when collaborating: writing, drawing and manipulating objects on a flat surface. People often rapidly transition back and forth between these actions. Pen computing devices allow fluid transition among these activities by using one medium to perform them all – a pen. Therefore, the use of Tablet PCs is advantageous. This is in contrast to existing computer-aided planning tools, which require transitioning between a keyboard and mouse to achieve an equal effect.

Tablet PCs in AgilePlanner are used as personal workspaces for story editing. They offer the capability of taking human handwriting inputs and converting them into equivalent digital (textual) representation. The employment of Tablet PCs gives AgilePlanner a conceptual edge over traditional computer-aided planning systems as information can be generated with ease using a pen, while the information is transported and persisted to the backend server readily available for remote and distributed access.

With respect to support for fluid transition between individual and collaborative work in agile planning, AgilePlanner implements a private and a public workspace with a synchronization mechanism that is transparent to users. Thus, the users can conveniently transition between two workspaces for personal and group work. AgilePlanner also supports transitioning between tabletop collaboration and work prepared externally by employing Tablet PCs. It is recommended that all team members have a Tablet PC of her/his own when using AgilePlanner. In this way, external work can be prepared and shared among planning participants, if need be, using Tablet PCs.
In addition, the use of the digital table and the Tablet PCs provides flexible user arrangements, allowing the user to engage various collaboration styles as they see fit. For example, physical placement of users becomes less of an issue, as planners are able to collaborate while standing in a corner (see Figure 5.1) or sitting around the table (see Figure 4.2). AgilePlanner provides shared access to digital objects on the iLab table. Thus fulfills the last interaction-related requirements previously identified.

![Figure 5.1: Agile planners engage in collaboration using a corner of the iLab table and a Tablet PC.](image)

5.2 Limitations

AgilePlanner provides partial support for simultaneous user interactions, partially because concurrent multi-user input on the iLab table is limited. The iLab table and the iLab table software can support up to two concurrent user inputs. In addition, my late discovery to this concurrent user input feature also contributes to the result that no concurrent user input support is enabled on the iLab table. The impact of this deficiency is noticeable in many planning sessions using AgilePlanner. People intercept other peoples’ interactions with the table by unconsciously making physical contact with the table while it is in use. The cursor erratically jumps between the two touch points,
competing for system access. When this happens, people resort to oral communications and gestures to resolve conflicts. According to my observations, the limited input capability does not seem to frustrate people much as conflicts could be easily resolved. Another criterion that AgilePlanner does not support due to pure hardware limitations is the requirement for supporting “use of physical planning objects”. The iLab table does not support any reporting mechanism that allows software to recognize the occupying objects, and differentiate them from input touch points.

Despite AgilePlanner’s capabilities facilitating the identified tool criteria, its instability made it less dependable than the team’s traditional planning tools. System breakdowns often occurred during information organization on tabletop. The breakdowns were caused when a high volume of Web Service traffic was generated between AgileTable and MASE for requesting as well as confirming story persistence, updates notifications and coordination. One of the possible reasons that cause slow communication process can be because the fact that Web Services may suffer from poor performance due utilizing text-based messaging infrastructure [WIKIPEDIA: Web services 2005], as the backbone of Web Services, the XML document model, “explicitly does not count among its design goals either conciseness of encoding or efficiency of parsing (service information)” [WIKIPEDIA: XML 2005]. In addition, AgilePlanner is a research prototype that incorporates diverse new technologies. Therefore, its design and implementation was my best-effort result. However, many improvements could still be made on them. Another major setback on system stability is that it cannot handle application transactions with grace. Since messaging with Web Services induces high time latency, AgilePlanner always employs asynchronous service invocations to shield its
planning interface from hold-ups resulting from the backend communication. When errors occur and there may be uncompleted requests queuing for service, the current implementation does not have the capability to undo front-end changes nor recommit the requests to MASE. Therefore, it may happen that the plan displayed on the tabletop is out of sync with the one on the MASE server without planners’ knowledge until the communication exceptions are programmatically detected, generated and caught.

5.3 Summary
The result of this qualitative analysis indicates AgilePlanner’s is a proof-of-concept tool that incorporates UB concept and agile planning. Conceptually, the tool should facilitate face-to-face collaboration during iteration planning. In the next chapter I present an empirical analysis of AgilePlanner. This is done by applying the system in real planning sessions, and I will report findings regarding usefulness and usability of the system.
CHAPTER 6. EMPIRICAL EVALUATION

This research is set out to investigate the feasibility of tool support to mediate face-to-face collaboration in agile planning as one unified and integrated process. In Chapter 5, I concluded that AgilePlanner is indeed a first-of-its-kind tool that is capable of combining capabilities of paper- and electronic-based planning media by fulfilling most of the necessary requirements identified in Chapter 3. In this chapter I evaluate its value in practice.

The empirical evaluation involves a preliminary case study and a follow-up interview. First, I introduce the objectives of the case study as well as the interviews. Then, I detail the experimental context, the instruments, and the data collection and analysis strategies used in this study. This is followed by my observations and their interpretations. Finally, I close this chapter by inspecting the limitations of this empirical evaluation.

6.1 Objectives

This empirical evaluation intends to achieve the following three goals:

1. to investigate if AgilePlanner is useful for iteration planning;
2. to understand if AgilePlanner is useable by planning participants; and
3. to assess if AgilePlanner can be used as a replacement to existing planning support

Understanding the concept of usefulness and usability of tools is fundamental to this evaluation. Usefulness is an equivalent term to utility [WIKIPEDIA: Utility 2005], which is defined as “the quality of being of practical use” by [Cognitive Science Laboratory]
1998]. For a tool to be useful for a work process, it has to provide a set of functionalities and make sure its generated outcome is applicable to the process. In other words, tool utility emphasizes on completion of tasks and quality of the resultant outcomes that a tool is devised to facilitate.

“Usability is a term used to denote the ease with which people can employ a particular tool in order to achieve a particular goal” [WIKIPEDIA: Usability 2005]. The notion primarily concerns with tool users’ psychology and physiology in mind when using a tool by asking questions, for example: Does it take less time to accomplish a particular task? Is it easier to learn and more satisfying to use the tool? Equivalently put, tool usability focuses more on user perceptions during tasks completion process. Measurements of user perceptions can be both objective and subjective. Objective measurements can include time that it takes people to complete basic tasks and a number of mistakes people make when using a tool, etc. Subjective measurements can include whether or not people feel a new tool is more efficient to use, easier to learn, or satisfying to use, etc than the existing tools. The assessment usually consists of multiple test sessions where the results of the first test are used as baseline for all the subsequent tests to compare with [WIKIPEDIA: Usability testing 2005].

6.2 Experimental Context & Participants

The evaluation process lasted for four months from May to September, 2005. During this period of time, the team worked on two independent projects, in parallel, in an academic environment. Both projects are developing research prototypes built to realize abstract concepts. The team was composed of four members: three of them are developers and
one is the on-site customer/project manager. I was also working in the same location, but did not participate in any development work.

The three developers had prior development experience ranging from several months to a couple of years. The discrepancy of development experience seemed to carry little significance to individual’s planning performance during the course of the evaluation. None of the developers possessed all of the skills that were required for either project. There was lots of individual research and peer learning during the timeframe. Also, the complexity of at least one of the projects was beyond the developers’ grasp initially. Many ideas, suggestions and discussions were exchanged among the team both within and outside of iteration planning sessions.

The empirical evaluation involves one case study on one agile team. The evaluation has two experimental steps. The first step of the investigation observes the team’s original planning practices regarding tool usage. The second step examines the team’s interaction with AgilePlanner during iteration planning. Over the four-month timeframe, the team had four iterations. The 1st iteration planning session was monitored for the first experimental step, and the 2nd, 3rd, and 4th iteration planning sessions were monitored for the second step. The physical settings for the two experimental steps were different. An interview was conducted with each team members after the three sessions of use of AgilePlanner. Both the case study and the interview were approved by the ethics committee at the University of Calgary, and the application form as well as the approval letter can be found in Appendix B and C.
6.3 Instruments
The first experimental step was completed with the 1st iteration planning session. In this session the team exercised their original planning practice. The practice involved heterogeneous planning media including both paper index cards and MASE. Some stories were created directly with MASE, others were created with index cards and entered into the system afterwards. In the subsequent three iteration planning sessions the team used AgilePlanner. Before their first time using AgilePlanner, a brief (less than ten minute) explanation of AgilePlanner was given to the team members.

6.4 Data Collection and Analysis
Both experimental steps in the case study involve observations, videotaping and post-analysis of the tapes. The population of the participating agile team is four people. The analysis concentrates on seven evaluation criteria described in the next section and the results are anticipated to be preliminary. A follow-up, individual interview is given afterwards, and the results are interleaved with the results of case study.

6.4.1 Evaluation Criteria
The first step to analyze the tool utility and usability for agile planning is to identify facilitating tasks (based on the definition of utility and usability stated above). I break down the iteration planning process into a sequence of activities that tools should assist:

1. **Review and update the previous iteration plan.** At the beginning of planning, developers report their progress on stories they were assigned in the previous iteration, and bring up any issues and problems facing them during the iteration.
The review usually starts by first identifying completed stories. The identified stories will be marked as completed in the database so that they will not appear in future planning sessions. The unfinished stories are placed in the Product Backlog. This act signifies that these stories should be tended in the future according to the team’s emerging needs. The review and update process usually ends with marking the past iteration completed from the project. Tools can facilitate the completion of stories and the transition of stories from the past iteration into the Product Backlog.

2. **Create a new iteration.** A new iteration is created. The target date of the iteration is always determined in planning. The usual length of an iteration is about 2-4 weeks; however, it is adjustable according to team’s release strategy. Also discussed at this point is the scope of development in the new iteration, any specific emphasis the next release has (if any), and committable employee work hours for the new iteration. Tools can facilitate communication by allowing relevant information from different sources to be easily available for all the participants.

3. **Story creation.** The onsite customer(s) create new stories based on the scope of the iteration and the needs of stakeholders. These stories are open to some negotiation. The stories in the Product Backlog are also reviewed to see if any of them fit into the current development priorities and scope. Tools can assist in the creation and modification of the story content by making the process faster and easier.
4. **Face-to-face interaction.** The customers explain what each story means and detail the desired effects resulting from the completion of the story. The developers respond with attempts to further clarify what the customer really needs by projecting effects from alternative solutions. During this phase tools are primarily used to organize story information or facilitate exploration of the story’s content in order to mediate the interaction process.

5. **Story estimation.** Once the content of the story is clearly understood by the developers, they start puzzling out estimated work hours for its completion. Tools can assist estimation by making it easy to undo/redo estimation multiple times. In addition, when a story estimation is too large, the tools should allow a story to be broken down into multiple subtasks. Tools can help by keeping track of the sum of all estimates for the upcoming iteration.

6. **Story prioritization.** It is typical that the work estimate of the desired stories exceeds the committable work hours of the current iteration. As a result the customer(s) need to prioritize stories by leaving less urgent features to the future. Tools can assist by highlighting the difference between the available and requested work hours to the meeting participants.

7. **Story persistence and remote access.** Planning usually closes by persisting the plan. The needs for stories to be electronically persisted and remotely accessed are simply inevitable when some project participants work partially from off-campus. Tools can mediate the persistence process by making it more efficient and effective, as well as enable remote access.
6.5 Interview

The four interviews have been transcribed and are available in Appendix A. These interviews are given after the agile team has used both their traditional planning tools and AgilePlanner. The aim is to collect user perception on both planning practices of using their original planning tools and AgilePlanner. The interview questions are designed to be both explorative and explanatory. Explanatory interviews aim at making explanatory claims about the population [Wohlin et al. 1999]. Explorative interviews are used as a pre-study to a more thorough investigation to assure that important issues are not overseen [Wohlin et al. 1999]. The interview examines the relationships between existing planning tools and AgilePlanner as well as several explanatory variables. As such, the results may allow me to explain why the planners prefer one tool over the other if such preference exists. The explorative aspect of the interview targets at creating understanding on how the planners perceive existing planning support and issues associated with using them. The interpretations of the interviews are integrated with the post-analysis of the case study, which is presented in Chapter 6.6.5 – Interpretation.

6.6 Case Study

“A case study is conducted to investigate a single entity of phenomenon with a specific time space ” [Wohlin et al. 1999]. An advantage of case studies is that they are easier to plan but the disadvantages are that the results are difficult to generalize and harder to interpret to situations that are different from the sampled case [Wohlin et al. 1999].

My case study monitors the agile team’s planning practice regarding tool usage during the two experimental steps. In the first step, I observe the team’s original planning
practices regarding tool usage. The second step, I examine the team’s interaction with AgilePlanner during iteration planning.

### 6.6.1 Step One: Original Planning Practices

The goal of this experiment step is

- to elicit usefulness and usability issues associated with the team’s original tools;
- to form a baseline for comparison for AgilePlanner.

Figure 6.1 shows the pictorial representation of the physical setting. “P/C” stands for a rear-projected computer where MASE was running, “T” stands for physical table, and “S” illustrates the seating area. The “P/C” is equipped with wireless mouse and keyboard for system interactions. Figure 6.2 illustrates the real planning setting for the first experiment step.

![Figure 6.1: Pictorial representation of the physical setting of the first experimental step.](image)
Figure 6.2: The real planning setting for the first experiment step

6.6.2 Observations

Observations are made in terms of the seven criteria identified above. In terms of the team’s original tool support for **review and update the previous iteration plan**, information projected on “P/C” (see Figure 6.1) was hard to read and the mouse movement was difficult to follow for people sitting in the area of “S”. The team members resorted to finger pointing, usually in combination with verbal descriptions, to help each other locate the stories under discussions on the screen. Alternatively, they used the mouse to highlight the stories; usually only the owner of the input device was able to keep track of the mouse’s movement. It is not a very effective approach as it is not easy
to keep track of a cursor fast moving across a large screen. As the team proceeded to
mark stories or iterations complete, or placed unfinished stories under the Product
Backlog, it is observed that only the person, usually the project manager who performs
these tasks, knew exactly what was happening on the screen due to the cursor issue.

With respect to tool’s mediation for creating a new iteration, the creation task needs
to be preformed in two places; one on the physical table and the other on MASE. The
task division induced waiting. To avoid it, sometimes the project manager would create a
new iteration on MASE in advance.

Story creation and face-to-face interaction usually goes hand in hand with one
another. The use of physical planning media allows for easy face-to-face communication
since everyone is facing toward to the new story cards placed at the center of the table.
Hence everyone’s reactions could be easily detected by one another. In addition, foreign
objects on the table had been seen to be used occasionally as means for elaboration. For
example, cups and beverages cans were observed one time being used to represent
distributed application servers and a developer used a pen and moving it among the
foreign objects to illustrate communication path and interaction sequence among the
represented servers. Also, story creation with index card is very fast, and the cards were
quickly organized and re-organized as the team’s needs emerge. Overall, the convenience
of paper index cards and their easy manoeuvrability sustains story creation and
communication activities quite successfully.

The developers simply grabbed the stories assigned to them and attributed story
estimation using their pens on the index cards.
Story prioritization started with a summation process of the estimates of all the stories in the new iteration. It was observed that the team added up all the figures manually, sometimes with the help of a calculator. As stories got moved in and out the new iteration, corrections to the previous summation result were applied otherwise the summation process was repeated.

6.6.3 Step Two: Planning Practices using AgilePlanner

The goal of this experiment step is

- to see how people interact with AgilePlanner

- to compare user interaction with AgilePlanner to their original planning media.

Through this experimental step I elicited usefulness and usability issues associated with the team’s practice with AgilePlanner. Figure 6.3 shows the physical setting of the second experimental step. In the figure, “D/T” stands for the digital tabletop – the iLab table. The tabletop can accommodate six adults comfortably in the seats labelled as “S” in the figure. We had two Tablet PCs, labelled as “TP” in the figure, during the experiments; although one Tablet PC for each planning participant would be the ideal. The Tablet PCs were shared among the four team members for planning activities. Figure 6.4 demonstrates the real planning setting for the second experiment step.
Figure 6.3: Pictorial representation of the physical setting of the second experimental step

Figure 6.4: The real planning setting for the second experimental step
6.6.4 Observations

AgilePlanner displays the last iteration plan as it was last seen to facilitate the review and update of the previous iteration plan. AgilePlanner preserves previous planning setting. By this feature, I assume the system may remind the team of the rational behind the arrangement of stories. This feature was observed to enhance and extend project memory. With the iLab table it is easy to engage face-to-face interaction in this environment as people are sitting almost right in front of each other, as shown in Figure 6.3. However, removing completed stories and iterations is not supported by the current implementation of the interface of AgileTable. It can only be achieved by firing equivalent commands on the Web end of AgilePlanner and then re-launch the table interface. Creating a new iteration is also not supported.

Story creation and face-to-face interaction usually go hand in hand. Two observations were extracted. First of all, AgileTablet took long time to get ready for operation compared to using pen and paper for story creation. The wait time seemed to put a halt on the flow of collaboration; as a result, some team members showed signs of impatience and being bored. Secondly, since no peripheral space is provided to hold foreign objects on the iLab table and with single user input support, any objects placed on the tabletop would compete for system access to AgilePlanner with the real cursor; which interferes with planning. Consequently, people had to write on the Tablet PCs held in their arms. Clearly, this is a rather limited and uncomfortable work posture compared to writing on paper index cards on the physical table. Lastly, people occasionally used paper-based media such as note pads and scarp papers for the elaboration purpose despite the presence of the Tablet PCs.
After stories were created, AgileTable displayed them on the tabletop (“D/T” in Figure 6.3). Four differences were observed in interacting with the displayed stories, in comparison to manipulating paper index cards on the top of a real table. First, since AgileTable encodes stories with visual characteristics, the team could quickly tell the type of the stories even when the story content was too far to be read clearly. Secondly, using AgileTable, an image card can be resized to increase its readability from all viewing angles. Moreover, in one occasion it was observed that the customer placed two enlarged story cards side by side to explain how they were different as features – resizing for the purpose of communication. Thirdly, editing the content of a story with AgileTable is very hard because the Story Editor (see Chapter 4.1.3) can only take inputs and display information from one fixed orientation on the tabletop. Thus, the feature was of little use during the course of the study. Lastly, people tended to forget that AgileTable does not support concurrent user inputs even when it had been made clear at multiple occasions, yet from time to time they still intercepted other people’s interaction with the table.

In terms of AgilePlanner’s support to **story estimation**, it is nearly identical as how index cards are used to fulfill this criterion. AgilePlanner mediates **story prioritization** by automatically updating the sum of the estimate of all stories assigned to the new iteration. This feature saved the team some time as they did not have to re-calculate the figure manually. The rest of the observations on using AgilePlanner to fulfill story prioritization are identical with the ones that have been recorded for face-to-face interaction mentioned above.

The team did not need any post-planning activities to enable **story persistence and remote access** as AgilePlanner and MASE had taken care of that along with planning
itself. However, the team did feel the need to review the planning results using MASE to make certain that the content was correctly translated, as handwriting recognition cannot be 100% accurate. The plans were correct for the most part.

### 6.6.5 Interpretation

#### 6.6.5.1 Usefulness

<table>
<thead>
<tr>
<th>Evaluation criteria \ Tool</th>
<th>Better Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review and update the previous iteration plan</td>
<td>Agile Planner</td>
</tr>
<tr>
<td>Creating a new iteration</td>
<td>Tied</td>
</tr>
<tr>
<td>Story creation</td>
<td>Original Planning Tools</td>
</tr>
<tr>
<td>Face-to-face interaction</td>
<td>Original Planning Tools</td>
</tr>
<tr>
<td>Story estimation</td>
<td>Original Planning Tools</td>
</tr>
<tr>
<td>Story prioritization</td>
<td>Tied</td>
</tr>
<tr>
<td>Story persistence and remote access</td>
<td>Agile Planner</td>
</tr>
</tbody>
</table>

**Table 6.1: Summary for usefulness comparison between Agile Planner and the team’s original planning tools**

Through observations of the two experimental steps and the post-analysis on the captured videos, I evaluate the utility of Agile Planner and the team’s original planning tools against the previously identified evaluation criteria. Table 6.1 summarizes the results and allows me to conclude that Agile Planner is slightly less useful than the original planning tools. The rows in the table that are highlighted in blue are areas that Agile Planner excels over the team’s original planning tools; the ones in white represent a tie; the ones in orange represent inferior aspects.

To the tying aspects, both Agile Planner and the original tools require more than one tool interactions to complete iteration creation. With Agile Planner it is an issue that can
be solved by improving its user interface design for instance, to allow a new iteration
to be part of the planning interface immediately after it is created without needing the
program to restart. On the contrary, with original planning tools there is no obvious
improvement can be done to create a new iteration on both the tabletop and MASE with
an integrated action because that would require a direct means of communication
between a physical table and a digital system which does not exist. As for story
prioritization AgilePlanner provides less effective support since it does not allow
concurent tabletop interaction. However, physical planning media lack computation
capabilities. Thus, when a story’s iteration assignment is changed, a manual calculation
need to be performed.

In terms of excelling areas for AgilePlanner, first the team usually planned with a
large vertical screen to facilitate equal information visibility using their original planning
tools. However, such setup induces difficulties for the users to specify a screen object
because the planners need to perform mental translation to match the object being
described or pointed at and its image representation on the screen. In addition, plan
readability deteriorates as the distance between “P/C” and “S” (see Figure 6.1) increases,
and such distance is not always easily adjustable. Even if sitting in close range to the
screen, it is still hard to track the mouse movement unless you are controlling it.
Secondly, using paper index cards requires entering stories into MASE after the meeting
has ended. This effort can and should be avoided.

With respect to the inferior areas for AgilePlanner, they are by and large subject to
tabletop interaction. Although AgilePlanner attempts to render naturalness to its tabletop
interaction and succeeds in producing a simulated environment by employing advanced
HCI technologies, the interaction it renders is still less efficient and convenient than physical planning media.

“Index cards are a little bit quicker right now. If the application was a lot quicker, even just exactly the same, I’d use it definitely. I’d use it full time if it was quicker routinely. Efficiency is especially important for big meetings. If such software exists it would be cool to show it off too. The thing is that it would have to be almost identical to the exact process of using the paper index cards.” – interviewee 2, answer 9

Also,

“Index cards have some things that you can’t replace with a computer, for example, their weight is a few grams. They are very easy to use, you only need a pencil. You do not have lots of setup.” – interviewee 3, answer 13

6.6.5.2 Usability

Some usability issues of AgilePlanner are caused by distributed communication and software bugs. System breakdowns happened multiple times in one planning session and intervention from the application developer was required. Due to the system’s incapability to gracefully handle service request transactions, the planners could be still planning actively with the front-end while errors had taken place behind the scene. When the exceptions were trapped, some planning activities needed to be redone and considerations needed to be recalled to get back to the latest planning state. This created stress on the planners. These issues were eliminated by running both AgileTable and MASE on the same machine as the element of distributed communication was stripped down to minimal.
Some usability issues are a result of missing features from the software implementation. For instance,

“It kind of defeated the purpose of having that collaborative workspace, where you can rotate the cards and look at the cards as if they were on the table. But as soon as you try to look at the details of the cards, your ability to interact with that card is significantly hindered by your selection of a seat at the table.” – interviewee 4, answer 14

This is due to the Story Editor on the iLab table supports data entry with only “straight-up” orientation. Also,

“the disadvantage entering information with Tablet PCs is a bit inconvenient. I think if one could make writing directly on the tabletop then it would be better.” – interviewee 1, answer 5

In addition, some user complained about AgilePlanner not providing the capability of allowing dynamic planning semantics associations to certain area on the tabletop.

“The problem was, the system didn't know what kind of semantics we had in a certain area. The system wasn't able to automatically translate that into something internally – somebody else had to transcribe the meaning.” – interview 3, answer 12

Some usability issues are as a result of hardware limitations. For instance,

“The other inconvenience is that the iLab table simply doesn't support entering information well. So we were back to using Tablet PCs or MASE for adding additional cards. ... What we saw in this table was primarily that it was designed based on the notion of how people are using tables - physical tables. They focus more on tabletop objects manipulation and less on entering of information. In agile planning, both parts –
organising as well as entering information – have the same importance. You need to
be able to do both. If you only support one side and not the other, basically, it's not
usable.” – interviewee 3, answer 12

Besides, the iLab table does not have space on the tabletop to place foreign objects.

“For example, we're sitting at iLab table, and there's no place to put your tablet.
You have to hold it in your lap. If you're sitting at a table, you can put your laptop on the
table, and still use the paper and pen not affect the communication between your team
members. I think that was the big, big disadvantage there.” – interviewee 4, answer 16

Also, Tablet PCs, although sustaining the handwriting and text recognition
capabilities, are not yet a competition to paper-based media in terms of convenience,
efficiency and effectiveness when it comes to tabletop facilitation.

“The technological costs that are incurred by the index cards are much lower.
Writing is learned in first grade. Index cards weight a few grams and laptops weight at
least 2 pounds. Apparently, carrying index cards around is much easier than laptops.
The start-up time of a pen is 2 seconds - I take off the cap, and that’s it. The start-up time
for a computer, connecting it to a Web server, is at least tens of seconds. Simply the
convenience factor is quite different.” – interviewee 3, answer 8

In spite of utility and usability issues, the tool concept that AgilePlanner represents
seems promising. Overall, people like the tool concept that AgilePlanner represents, and
provide positive feedback and constructive criticism toward the tool such as

“I think it is an intriguing idea. It facilitates iteration planning tasks to be
completed at once. It’s like a toy and you feel like playing with it because the interface is
colourful. It is more appealing to my visual. Also, objects are shaped differently and
different events triggered different responses, so it makes me feel more engaging and participating in planning. Unlike pens, paper and tables, you know they are dead objects and they do not appeal to your attention, at least not actively.” – interviewee 1, answer 5

“I am actually kind of liking it. It’s very close to actual using paper index cards to do agile planning. Not only the technology is very cool, so it actually keeps your interest in it. Just being able to handwrite on the card as we do normally makes it a little bit quicker. You have a nice easy visual display of front too. I think it is pretty cool. I think it’s a really cool idea, just make it feel a lot more like a real iteration planning meeting.” – interviewee 2, answer 7

On top of the “made” intuitiveness, the tool concept makes room for planning enhancement possible with computing power - which is infeasible for physical planning media. In addition, using AgilePlanner allows planning to be done at once without switching data storage media. Such advantage is an asset to agile planning considering the following scenario:

“You may have several index cards that mean similar things as you have so many ideas flying around all the times. Being able to find those similar ones and filing them together into MASE takes a bit of time. I wouldn’t want to be having to enter them after the planning session. I’d rather being able to do it right then and there.” – interviewee 2, answer 6
6.7 Limitation

The case study and the interviews shed light on the effects of the tool concept realized by AgilePlanner. Although the derived insights are valuable, yet there remain the following threats to their validity.

First, due to the time and resource constraint of the M.Sc. study, the empirical evaluation was only run over a short term and with limited samples that might not be representative of the target population. Thus, the interpreted results could suffer from this shortage of realism in the experiment settings.

Second, the experiment participants belong to the same research group as the observer; which may introduce a positive bias into the interview results.

Third, the case studies and the interview are conducted and analysed by the same investigator – who also is the developer of AgilePlanner. This inevitably introduces a research bias to the interpretation of the collected results. Precaution was applied to mitigate the impacts of such bias during analysis process.

Fourth, all the empirical treatments are conducted on the same agile team. As such, the number of perspectives is quite limited and the scope is narrow. Results generalized based on the small size sample population may be questionable by nature.

Fifth, no similar empirical evaluation is done in industrial settings. Developers from academic and industrial settings possess different perspectives and the experiment findings could potentially not reflect industrial users.

Sixth, the agile team under evaluation has an inherent interest in the advanced technologies, digital tabletop displays and Tablet PCs involved in this research project. This potentially contributes user bias to the evaluation results.
6.8 Summary

In this chapter I have shown the methodology and results of a case study and interviews on AgilePlanner. The results of this empirical evaluation indicate that the concept of the tool, although not yet as fast and convenient as physical planning media, is a close enough electronic planning medium and has perceived benefits of highlighting, organizing story information with freedom, encouraging collaboration by visual effects, and preventing story duplications on different storage media. Given the exploratory nature of this empirical evaluation, the findings are encouraging enough to suggest more advanced implementations of the tool concept should be constructed, and more thorough investigations should be done on them in the future.
CHAPTER 7. CONCLUSION AND FUTURE WORK

In this chapter, I summarize my research contributions. First, I revisit the thesis problems that motivated this work. Then I describe my research contributions by showing how these problems have been addressed. I close by suggesting areas of potential future work.

7.1 Thesis Problems

In Chapter 1, I presented two research questions regarding effects of incorporating UC into agile planning.

- **It is unknown what the requirements should be for a tabletop-based computer tool to facilitate co-located face-to-face collaboration for agile teams during iteration planning.** In software engineering, prior research on computer-aided tool support has focused on maximizing performance of individual work using single user computers. When it comes to collective activities, the use of single user computers is just out of the place. Prior work on UC has focused on how to manipulate the embedded information in an environment. There is limited research on the effects of using UC tools to mediate collective agile development activities. Consequently, requirements of UC tools for agile planning are unknown.

- **If such a tool support is built, it is unknown whether or not it will be useful and usable.** In case a novel tool is needed and after such tool is designed and implemented, it is important to evaluate its usefulness and usability for iteration planning.
7.2 Thesis Contributions

This thesis addresses the problems mentioned above with the following research contributions:

1. proposal for introducing the UC concept into the agile planning process (problem 1);
2. identification of seventeen criteria essential for the proposed tool support (problem 1);
3. an evaluation of existing agile planning tools against the identified tool criteria (problem 2);
4. the design and implementation of AgilePlanner, a tool that supports the proposed tool concept (problem 2) and
5. an empirical evaluation on how AgilePlanner can be used by agile teams and its effectiveness to facilitate agile planning (problem 2).

These contributions are elaborated below.

I addressed the 1st thesis problem by first proposing to incorporate the concept of UC into the agile planning process. To this end, I analysed the usage of existing planning tools. The use of pure physical planning media, although effective for face-to-face interaction, does not support agile project management activities. The use of computer-aided agile project management systems offers the complementary features to the use of physical media. The combined use of them induces costs of data conversation and possible data replication in both physical and computer worlds.

UC promotes embedding information in everyday objects so that it can be available everywhere. Tools built based on such a concept allow users to concentrate on the tasks
at hand rather than on the tools. Typically, physical tables are used for agile planning. The only digital table available for this research was the iLab table. It was noticed earlier in this research that the hardware device is useful for information display and organization, but not information input. Thus the use of Tablet PCs is determined.

Then, I addressed the 1st thesis problem by identifying seventeen tool criteria essential for the table-top based computer tools to assist agile planning. The identified criteria consist of functional requirements facilitating agile planning activities as well as agile project management, and interaction-related requirements mediating planning collaboration. Based on these tool criteria, I evaluated existing planning tools.

This addressed the first part of the 2nd thesis problem. The evaluation results showed no existing planning tools fulfill the identified criteria in full. In other words, they are inadequate for agile planning as they either facilitate agile planning activities or planning collaboration, but not both. As such, there is a need for a novel tool that attempts to provide all the identified capabilities. As a result, I designed and implemented AgilePlanner, and confirmed that it satisfies the proposed tool concept; which addresses the second part of the 2nd thesis problem. This was done by conducting a qualitative assessment of AgilePlanner against the identified criteria. Lastly, I empirically evaluated AgilePlanner for its effectiveness by conducting a case study on an agile team. In addition follow-up interviews were conducted with the team members; this addresses the third part of the 2nd thesis problem. The experiment monitored four iteration planning sessions. In the first iteration planning session, the team planned with their original planning tools. For the rest three sessions, they planned using AgilePlanner. I outlined seven criteria that are essential for computer support to be useful
for agile planning. As summarized by Table 6.1 AgilePlanner is slightly less useful than the team’s original planning tools mainly because of the hardware limitations in relation to physical planning media. For example, using index cards is more effective and convenient than using Tablet PCs because writing on paper does not require start-up time and carrying index cards around is very easy. AgilePlanner has many usability issues. Some result from imperfect software design and others are as a result of hardware limitations. For instance, AgilePlanner could have facilitated the functionality “create iteration” in a way that would not interfered with tabletop planning activities. In addition, it would be more close to physical tabletop interaction had entering information right on the tabletop been adequately facilitated by the iLab table. However, the fact that the table does not mediate entering information lets us employ Tablet PCs for adding stories; this makes the whole setup a bit less natural to work with. As such, AgilePlanner is not a ready replacement for existing planning tools. However, it is worth of noting that the results of the interviews revealed positive user feedback toward the tool concept underpinning AgilePlanner as manifested by the following quote:

7.3 Future Work

The work presented in this thesis is only a preliminary step in constructing effective tool support for collaborative planning in an agile software development environment. There remains lots of room in this research area for future work.

AgilePlanner was a research prototype tool with defects and system instabilities. It will be beneficial to improve the design of the tool in terms of performance and usability, in order to better facilitate face-to-face interaction and team collaboration.
In terms of hardware capabilities, a digital table that possess more properties of a real table is needed. Although the iLab table is the state-of-art tabletop technology, it possesses limitations to be used as a real table. The tabletop display does not provide space for laptops or tablets, this hampers the convenience factor of using a table. In addition, currently the iLab table allows maximum two concurrent user inputs. This is limited considering the typical numbers of people can use a tabletop at the same time. Also, higher input resolution will be needed to allow handwritten information inputs.

In terms of software interface design, it will be desirable to investigate new interaction techniques. For example, the visual characteristics implemented by AgilePlanner provide a subtle but effective means for content differentiation. Future work can look into multimedia techniques to offer effective means that reveal different aspects of information. As modern computers possess advanced capability managing multimedia resources, it would not be surprising that more interactive techniques will be invented to overcome the limitations of the physical world.

Lastly, the empirical evaluation has limitations in terms of how the case study and the follow-up interviews are conducted. The case study was limited in their observation timeframe, and experiment population. A longitudinal study will be more beneficial to determine the long-term effects of the proposed tool in evaluating the iteration planning process of agile software development teams. An experiment population that does not have a close tie with the researcher would be more helpful to derive natural user feedback through the follow-up interviews.
REFERENCES


APPENDIX A: INTERVIEW TRANSCRIPTIONS

B.1

Q 1: What do you think the advantages and disadvantages are using paper index cards?

A 1: In comparison to Tablet PCs, they are convenient to operate since the size is small and the weight is light. You can turn them around anytime and pass them to everyone. With Tablet PCs, you cannot see the display clearly from some angles. You cannot have share the content with all the team members. So, I think index cards have higher operability. They are easier to share, and assist in creating a smoother flow of interaction. The disadvantage is that they can be lost easily. If you create information on the paper but do not have permanent storage to preserve them, it means you can lose the information very easily. Then you cannot keep what you want.

Q 2: What do you think the advantages and disadvantages are using MASE kind of iteration planning systems?

A 2: The main advantage is that you can go back to the system for the reviewing or referring purpose at any time, as opposed to paper index cards if they are lost you can not recover those stories. For instance if you have a new story, and it is related to an old one, then you do not have a record of the old story if it is stored as index cards. You cannot recollect what you have done. However, with MASE you can search for it. Using MASE you can starch and better organize your project history. The disadvantage is that the screen of computers is usually not large enough for a group of people to interact with each other. Even in case of 3 to 4 people standing in front of a computer, operating that computer is very inconvenient.

Q 3: Could you please describe your previous iteration planning approach in terms of tool usage?

A 3: It starts with the customer coming up with new stories, and writing them down on paper index cards. Then these cards are placed on top of a table so that the customer can go through them and explain the meaning for each story. Then he will elaborate which ones are more urgent than others, then we developers pick up these tasks and estimate. In the end the project manager would appoint someone to enter the selected stories into MASE. The main inconvenience I have felt is that we cannot complete planning tasks at once. It would be better if the information could be entered into the system in parallel to the planning process, so that we can save some time. I don’t like that we need to work on one task with two different media.

Q 4: Does it make any difference for you in terms of how to interact with a planning system between handwriting and typing?
A 4: I still feel it is more convenient and natural; writing than typing. It really depends on individual’s habits. I just personally feel more comfortable writing. However, I do not feel either means would affect my interaction with my team members.

Q 5: What is your general impression about AgilePlanner?

A 5: I think it is an intriguing idea. It facilitates iteration planning tasks to be completed at once. It’s like a toy and you feel like playing with it because the interface is colourful. It is more appealing to my visual. Also, objects are shaped differently and different events triggered different responses, so it makes me feel more engaging and participating in planning. Unlike pens, paper and tables, you know they are dead objects and they do not appeal to your attention, at least not actively. The disadvantage is that entering information with Tablet PCs is a bit inconvenient. I think if one could make writing directly on the tabletop then it would be better.

Q 6: Do you think AgilePlanner can be a replacement for MASE?

A 6: AgilePlanner is better because every one has equal access to it at the same time. MASE can only allow one person to control the keyboard and mouse. And the turn-taking system interaction is bit troubling

Q 7: Do you think AgilePlanner can be a replacement for paper index cards?

A 7: I feel the two of them have a common disadvantage that is on a horizontal display one cannot see story details clearly. Added that sometimes stories are not placed in a upright angle, reading them stories is pretty harder. AgilePlanner has an edge over paper index cards in that you can complete, again, planning tasks at once; whereas if you want to preserve the story content on paper index cards, then you need to enter them afterward, and you have to repeat the same work twice. I feel between paper index cards and AgilePlanner, I would prefer AgilePlanner.
B.2

Q 1: What are the advantages and disadvantages using paper index cards for iteration planning?

A 1: One of the main advantages is as you are speaking, coming up ideas or anything that is related to the project, you can jug down information very quick, so then you do not have to record yourself, and to videotape to extract information later. So, you have it right there. They are easy to throw away. They are easy to edit, update and change. That is probably the main one. Also, all the index cards contain very brief information and you can expand upon them later, so they hold all the ideas. The main disadvantage I would say is the bookkeeping of them, because you may have many index cards at the end of the meeting. Sometimes you don’t jug enough information down you forget what it actually meant. Those are the ones that I can think of on top of my head.

Q 2: What are advantages and disadvantage using MASE kind of iteration planning tools?

A 2: I think the biggest advantages is being able to geographically distribute your agile team with different members being at different locations. You can actually all be part of an iteration planning meeting while being distributed. You do not have to be in the same office, as well too as cleaning up a lot of mess. You are working with actual software other than paper index cards; which do not have all the extra mess that comes with paper and ink. The biggest advantage is being able to distribute your team. The biggest restriction I guess is that it is not a natural process. You actually have to select fields and type in rather than using index cards – it is quick. You get to grab an index card and write on it. The biggest disadvantage is that using computer systems is not natural for the iteration planning process. Being agile you can use index cards since it is quick without having to go through selecting fields and typing the cards.

Q 3: Do you think the actual typing makes you feel unnatural about it?

A 3: I think basically there are people that are slower in typing. With many ideas that are actually being thrown out there at the same time, it would be difficult for them. You’d only work with one card at a time on the software rather than having a couple people working on different index cards at the same time. Each card contains very brief explanations for a story you come up with. I think it takes longer on the software rather than writing down by hands. Biggest disadvantage is just not as natural; you can’t quite get everything down as fast as you can with index cards.

Q 4: So, in the end in your team you basically use paper index cards first and then transfer in to MASE?

A 4: yeah

Q 5: In your opinions what is the rationale behind it?
A 5: Basically the last point, we can throw ideas out there very quickly makes the ideas going all the times, and paper index cards are very easy to write on. With software and with that many ideas, we can’t really catch up. So we all write them down on index cards. At the end, we go through the index cards and decide which ones are going to be using and then put them into the software, then we know what we actually have at the end of the planning game. After planning since we are not under constraint anymore, we can actually edit and put the story cards into the software. We can take an hour if we wanted to.

Q 6: I am just wondering if you have a large number of story cards generated in the end and then some people have to be the one who convert the information from paper into computer systems would you be willing to be the person that gets appointed to do that kind of job?

A 6: No it takes a lot more time. You may have several index cards that mean similar things as you have so many ideas flying around all the times. Being able to find those similar ones and filing them together into MASE takes a bit of time. I wouldn’t want to be having to enter them after the planning session. I’d rather being able to do it right then and there.

Q 7: What is your general impression about AgilePlanner?

A 7: I am actually kind of liking it. It’s very close to actual using paper index cards to do agile planning. Not only the technology is very cool, so it actually keeps your interest in it. Just being able to handwrite on the card as we do normally makes it a little bit quicker. You have a nice easy visual display of front too. I think it is pretty cool. I think it’s a really cool idea, just make it feel a lot more like a real iteration planning meeting.

Q 8: Do you think it can be a possible replacement for your current iteration planning tool?

A 8: I would hope that it would. Basically exactly the same except rather than using paper index cards. It’s quick enough; you can do more than one card at a time and you are not restricted to having only one person to create a new card at a time. You can discard digital cards like nothing with the software. I’d definitely use it.

Q 9: Do you think you can routinely use AgilePlanner in your iteration planning?

A 9: I’d probably think. Still I think the index cards are a little bit quicker right now. If the application was a lot quicker, even just exactly the same, I’d use it definitely. I’d use it full time if it was quicker routinely. Efficiency is especially important for big meetings. If such software exists it would be cool to show it off too. The thing is that it would have to be almost identical to the exact process of using the paper index cards.
Q 10: Using AgilePlanner in comparison to your previous iteration planning tools do you think the quality of the iteration between you and your team mates different?

A 10: No. Actually I find it to be pretty much exactly the same. We are all sitting around the table as would in a regular iteration planning meeting. We are all discussing because we have all the digital cards visually as you would with index cards. You see everything visually and everything in the past iteration, which a lot of time you don’t in real iteration planning meeting since you do not actually carry the old index cards. Here, you have. It’s visually right in front of you. You can easily carry story cards from the last iteration to a new one. So you know what is exactly there…yeah you are all sitting around the table it is exactly like a meeting. So you can discuss things and create index cards and move them around and do whatever you have to do.
B.3

Q 1: What do you think the advantages and disadvantages are of using index cards?

A 1: Index cards are cheap, fast, available, and very convenient. Everybody knows how to write, so there's basically no training. Planning with index cards keeps all the core information, and the cards are tangible - you have something that you can hold and pass around. The disadvantages are for tracking purposes, and there's only one instance of the index card. If you're not close by, then you can't actually look at who has what story, who's working on what, or about the current status. So, if you are distributed as a team, there is no way that you have a central location to actually keep track of what's going on in the team. Clearly, distributed planning meetings with index cards are not feasible. If you duplicate index cards, then you have all the problems with replicated data. Two cards can result in a situation when one is updated and the other is not. Then you cannot be sure who is working on the updated story, and it becomes difficult to keep tract of the effort that was spent on the story. Index cards are rather limited. They work nice with people in the same room, always are working at the same time, and sharing the state of the project as well as the iteration on the white board. Then the people can move the index cards to indicate "I'm working on it", "I'm done" and so on. For example, in our situation, where I'm not staying with you guys, and sometimes I'm working from home, and sometimes working on the road, when I'm travelling. I can't keep track of what's going on.

Q 2: OK, so how do you think the MASE iteration planning system helps in this kind of situation?

A 2: It gives me access. Assuming everybody enters all the information properly, and updating it timely, I'd be able to keep track of it from wherever I am as long as I have an Internet connection.

Q 3: Are there other advantages you can think of for MASE kind of planning system?

A 3: You could do distributed planning. That was the idea originally - to support distributed teams with project planning and focus-tracking. Somebody could enter a new story from another location, I refresh my Web browser, and I see it, and we can discuss the update. So in the end, it enables us to do distributed planning. Also the problem that I see is that entering information into a system like MASE is much more inconvenient than simply writing something on an index card.

Q 4: Right. Actually, one thing I was thinking, from a project planning point of view, how bad would it be if you cannot keep track of everybody's progress all the time?

A 4: That's highly inconvenient. Specifically, when you work with professional teams, you do not know what's going on. When I was away, it was highly inconvenient because you don't know what's going on, unless I have something like MASE.
Q 5: But just looking at MASE. The only information you get is the story card name, the estimate, and what sense of progress can you get by just looking at that table?

A 5: I'm not only getting that, I'm also getting if it's used correctly i.e. every evening people enter the time remaining for all the tasks. That information allows me to see the actual effort and the remaining effort. It allows me to see if the remaining effort is going down, and with actual effort I can see how much work is being spent. In principle this information – not only in MASE but in similar systems – can be used to create charts. Information is there. It's just a matter of creating the charts.

Q 6: how important is the information that MASE displays for a developer, in your opinions?

A 6: Basically, it's a kind of reminder that you have to work on specific things. It helps people keep track of what they are doing, and what they are supposed to be doing. It's like, there is a deadline, and deadlines help you focus your attention on specific things.

Q 7: Can you describe your current planning process as far as usage of planning media?

A 7: Oh, that's heterogeneous. Basically what we do is we have meetings, and we use either paper-based index cards, or enter information into MASE. If we use paper-based index cards, somebody has to enter the information after the meeting. So for keeping track of what's going on with the project, I'm trying to enforce the use of MASE, but I'm not very successful at it, at the moment, probably because we can simply talk to each other on a daily basis.

Q 8: Do you think the difference between typing and writing information affects meeting?

A 8: That's one of the issues. The other issue is that a Web page is different from a piece of paper. I can't carry a Web-based system with me everywhere. I can carry around index cards. The technological costs that are incurred by the index cards are much lower. Writing is learned in first grade. Index cards weight a few grams and laptops weight at least 2 kilos. Apparently, carrying index cards around is much easier than laptops. The start-up time of a pen is 2 seconds - I take off the cap, and that’s it. The start-up time for a computer, connecting it to a Web server, is at least tens of seconds. Simply the convenience factor is quite different. It's like reading a book - I can read a book everywhere - I simply have to carry it. In contrast, doing the same thing with a computer is much more inconvenient.

Q 9: so how does that inconvenience factor influence the quality of the collaboration through iteration planning?

A 9: That depends. That's actually, in my opinions, very strongly related to usability. We were using MASE during the planning meetings. It was slowing us down, because
somebody had to enter new user cards, and that took a little more time than simply creating an index card. Sharing 20 index cards in the team is very easy. Having everybody around a computer, although possible, is little bit harder. I don't think that it's not possible to use computer systems during planning, but the index card is still easier. Plus you have to switch between talking, communicating with the team members, and doing something on the computer. Quite personally, you type, and everybody else watches. So overall, I would say it's feasible to use computer systems during planning. We were using MASE in some of our planning meetings. However, index cards were faster. So at the moment, my tendency is still "let's do the planning meeting with index cards, and then enter the information we gather into the computer afterwards."

Q 10: So how many story cards do you create for one iteration? The reason I ask this question is that apparently you have to compensate the cost that in the end, if you have someone to enter information from paper into computer, you actually have to compensate that cost with the cost of actually using MASE with the iteration because you say it is slower.

A 10: Yeah, I would put it that way. Entering information can be under one person, waiting for the slowness of the electronic system is done by everybody in the planning meeting. So I still would argue it's probably cheaper to first use index cards to completely focus on planning, and afterwards for project tracking better to enter them into the system.

Q 11: When you refer to a story, one using a physical table, and one using a SmartBoard or a whiteboard-like device, do you think there's a difference when you refer to a story?

A 11: Yes. On the table, everybody can sit around and simply point to a story. On the Smart Board, you typically have one mouse pointer, and only one person can point. So it's a little bit tricky.

Q 12: So what's your general impression of AgilePlanner?

A 12: I think AgilePlanner is a neat idea. The good thing about AgilePlanner is that it showed us what the problems with the current tabletop planners are. The primary problem that I see is that the iLab table supports organising information quite nicely, but it doesn't support entering information at all. It was highly inconvenient how we had to enter information with the system. There were obviously usability problems - it was a first prototype, so that was to be expected - but usability problems, for example, adding another iteration while doing the planning, wasn't supported. There were instabilities in the system; clearly this didn't help in the planning task. Overall, what I really liked is that we were able to organise the information ourselves on the table, simply by placing information somewhere on the space, and then claiming these are the things to be deleted, these are going into the backlog, these are done even when it wasn't indicated anywhere on the screen. The problem was, the system didn't know what kind of semantics we had in certain area. The system wasn't able to automatically translate that into something internally – somebody else had to transcribe the meaning. The other inconvenience is that
the iLab table simply doesn't support entering information well. So we were back to using Tablet PCs or MASE for adding additional cards. There was not even a space around this table for putting your laptop around that. So I think what we saw in this table was primarily that it was designed based on the notion of how people are using tables - physical tables. They focus more on tabletop objects manipulation and less no entering of information. In agile planning, both parts – organising as well as entering information – have the same importance. You need to be able to do both. If you only support one side and not the other, basically, it's not usable.

**Q 13:** Do you think AgilePlanner can be a replacement for your current planning media?

**A 13:** Comparing with current planning media in the form of MASE, the answer is yes. In the end, I believe if we work out the bugs and the usability issues it will be more convenient to use AgilePlanner than the current table-oriented database and table interface that we have in MASE or even commercial tools. Having said that, the typical product state of AgilePlanner is leading me to believe I still prefer to do MASE-based planning. In the end, at the moment, if I have a co-located team, the highest preference is still on index cards, and then entering the information into a project tracking system. That results from usability issues. Index cards have some things that you can't replace with a computer, for example, their weight is a few grams. They are very easy to use, you only need a pencil. You do not have lots of setup. What I would say real interesting to test would be using AgilePlanner in a distributed setting, actually with 2 tables, and see if you can actually do a distributed planning meeting using the electronic equivalence of index cards.
B.4

Q 1: The first question is: What do you think the advantages and disadvantages with using paper index cards?

A 1: With paper index cards, it's easy to lose them. So it's a big disadvantage for the paper index cards. If you don't keep track of where you put them, you might lose them, and you might lose an important card. The advantage of them, though, is that they're really quick and easy to pull a card down, and write something quickly down, and put it on the table for everyone to see.

Q 2: Do you feel any obstacles in using paper index cards doing the planning?

A 2: Not really. The only obstacle I can think of is that sometimes people are holding the card in front of them and reading it, so you can't read that same card at the same time. When they leave them on the table, it's fine because you can read upside-down, but you can still see it. As soon as they pick it up off the table, you can't see it anymore.

Q 3: Do you find the space that the index card provides limited? How does the space influence your interaction with the card?

A 3: I haven't really experienced limitations when I've tried to use them, but I could see that if you're trying to cover a story that's really hard to explain, or really hard to write down, a note that would incite a question to continue a conversation afterwards. Sometimes there could be not enough room, or if you write down your comments and you have really big writing, the space could be more limited than if you have small writing, but I haven't come across that problem personally.

Q 4: What do you think is the advantage and disadvantage of using MASE?

A 4: The biggest advantage is that I don't lose anything or it's less likely that I'll lose a story. The disadvantage is that it's a lot harder to organise things into iterations, with a system like the MASE system, for example. If I wanted to set up priorities, it's really hard for me to create a stack of high-priority cards and medium-priority or low-priority, or whatever, because it sorts them alphabetically. I think it's limited that way. The other advantage is that it keeps track of your estimates so I don't have to have a calculator or add them up in my head.

Q 5: What kind of advantages does that give you to be able to organise story information during the planning with paper index cards?

A 5: It lets me have a better idea of how to manage my time, for example, there's 6 stories and 2 of them are really high priority, when they're stacked in piles of priority, it's really easy to say "these are the high priority ones, these are the ones I have to do first, I
have to get these done". The medium priority cards are the ones I'm going to try to get done, and the low-priority ones, well if I don't have time consideration, or something unexpected happens, they can always get pushed to the next iteration, and it's not a huge deal. It's a lot easier when you can visualise high-priority, low-priority, for example, or even in terms of people, because there's always somebody responsible, and in case something happens you can always just switch the people responsible a lot easier by moving the card. Using the computer system, if you're going to change who the responsible programmer is, you have to go in and physically change that, but if you're not using the computer system, you don't have to say "I'm not responsible for this anymore. You are." and I don't have to physically keep track of that. It's got a positive and negative to it.

**Q 6:** How do you feel about the form-based information presentation as used in MASE?

**A 6:** It's not really natural to go through it. They might be grouped together in a list, but they're still in a list. If I want to change something I have to go all the way down the list, whereas I can just cross something out on a paper card and just write something else. It's a lot quicker time-wise. I've got better things to do during the day, like getting the tasks done and the stories. I don't need to spend 3 hours in a planning meeting that I could do in 5 minutes with a pen and paper. It's more about efficiency than anything else.

**Q 7:** Can you describe what your current planning practices as far as tool usage goes?

**A 7:** Over the past 4 months, it's been a mixture of paper index-card based and computerised-based using our MASE system and the table-top system. It's been a mixture of the three. I can't say that one has been a dominant factor.

**Q 8:** When you are directly interacting with MASE through iteration planning, what's the disadvantage of using that kind of system?

**A 8:** hmm... what's a disadvantage... if we're working in groups of more than 2 people, it's hard for everybody to be able to interact with the system at the same time. On a 1-on-1 basis it's fine. I have my mouse, I have my keyboard, my chair, I can sit there and enter things, and set estimates and set actual values related to time. If there's 3 or 4 of us, which is usually the case, one person is always in the controlling position, with a really good view of the system, and can really interact, and everybody has to take turns going through that person to get their point across. And it's also an afterthought. You do your discussion as a group, and one person enters it into the system. So you don't get that interaction as you're planning, you interact first with each other, the group members and then with the system. So it's an afterthought. That's the biggest disadvantage. It's designed for an individual user to interact with it. You don't get the same speed and efficiency and interaction that you would use a paper system.
Q 9: With paper index cards you typically write, and with the computer system you typically type. Do you think these two different interaction techniques cause any difference when you interact with the planning media?

A 9: To some people, I think it does. For me, it doesn't really make a difference, just because I'm used to typing. What I do like about using a keyboard is that at the end of the day I can read what's written on the story, whereas I have trouble reading some people's handwriting. So for example, if I have a story card written by one of the other people in the lab, and their handwriting isn't very clear, I have to ask them to read and explain what they've written. Whereas when it's on the MASE system on the computer, I can read the text because it's in block letters - legible text. So I don't have to go to them to remind me what the cards are.

Q 10: one question that pops into my mind is: How often do you go back to MASE and check the stories that you entered in the past?

A 10: I'm usually there at least twice a day, just for entering time estimates and stuff. So it's always keeping it fresh in my mind. In terms of checking actually what the story that I'm supposed to be working on is, it's usually after I'm finished, then I go back and check the next one. So depending on the length of the story, a couple hours to a day or two. It really depends on what part I'm working on.

Q 11: So how does MASE serve you in the development process?

A 11: It's more of a reminder, or a time-tracking tool, right now as it stands, more than anything. That's because the current iterations are pretty small, in terms of stories to complete, so it's not much of a challenge for me to keep track of everything I have to do in my head. So I don't have to go back to MASE every two hours to remind what I'm doing. But I could see other people, or there's a whole bunch of small stories to be accomplished, people going back more frequently. I don't think I could give a numerical value of how often I go back. It's at least twice a day.

Q 12: You also mentioned earlier that when you use MASE it's kind of like unnatural, slow or inefficient in comparison to paper index cards and a pen. How does that kind of unnaturalness, delay, slowness. How does that play a factor to the iteration planning - the quality of the collaboration?

A 12: I don't know if that big of an affect on the quality of planning, it's just frustrating. I think the actual quality of the meetings is pretty much the same using paper, or using the MASE system. The meetings are small, and we don't have an outside customer - our main "customer" is internal and they're familiar with the system, it's more frustrating waiting for the system to load, and enter things. And while we're waiting to do that we're talking about the meeting. So it's not really affecting the quality of the meeting itself, it's just frustrating that the meeting takes longer than it has to. That's more my opinion of that, I don't think the quality is affect by using paper vs. computerised system.
**Q 13:** What's your general impression about Agile Planner?

**A 13:** I like it. Maybe I'm biased, but I like the idea that I can sit around facing the other members of my team and have a discussion, and interact with the computerised system at the same time. When we've been using MASE system in the past, one person always has to sit with their back to the rest of the group, and everybody's looking over their shoulder, whereas during the planning meetings with your system, it's more face-to-face contact, and it's a lot more relaxed than trying to type on a computer screen. I like the informality about it, and I don't feel as nervous with people looking over my shoulder. That's what I like about it.

**Q 14:** What do you not like about it?

**A 14:** The big thing is bugs. It was buggy but I understand that's part of the development process. I didn't like that when you double-clicked on cards, that if you were sitting at certain points of the table you couldn't rotate the card. It kind of defeated the purpose of having that collaborative workspace, where you can rotate the cards and look at the cards as if they were on the table. But as soon as you try to look at the details of the cards, your ability to interact with that card is significantly hindered by your selection of a seat at the table. Whereas when you're using the paper system, you can take that card right to where you're sitting, and write on it. I thought that was one of the biggest disadvantages of using the planner. Along with the interaction was similar to the table interaction, but as soon you had to enter in data, you went back to the point where your position in the meeting directly affected your ability to interact with the system.

**Q 15:** OK, so in your opinions, do you think AgilePlanner could replace your current planning tools?

**A 15:** Yeah, I think it can for organising a meeting. If all the stories are entered into the system, either by multiple individuals, or by a single individual, I think it would be a really good replacement for the current system. I think the only drawback that would prevent it from being implemented in place of, or overtop of the current system, is that you can't dynamically modify data once it's been entered, like changing an estimate, etc. However, if everybody's got a tablet sitting at the table, and everybody's able to interact with the system even that would be a lot better than the current digital system.

**Q 16:** So in your opinion, how would you feel the comparison between AgilePlanner and paper-and-pen are?

**A 16:** I think it's got a bit more to grow before it becomes a full replacement. It's still a lot quicker to do it by paper and pen. The technology itself is also part of it. For example, we're sitting at the iLab table, and there's no place to put your tablet. You have to hold it in your lap. If you're sitting at a table, you can put your laptop on the table, and still use
the paper and pen not affect the communication between your team members. I think that was the big, big disadvantage there.
APPENDIX B: ETHICS APPROVAL

MEMO

CONJOINT FACULTIES RESEARCH ETHICS BOARD

c/o Research Services
Room 602 Earth Science
Telephone: (403) 220-3782
Fax: (403) 289 0693
Email: plevans@ucalgary.ca

Wednesday, May 04, 2005

To: Lawrence Liu
Computer Science

From: Dr. Janice P. Dickin, Chair
Conjoint Faculties Research Ethics Board (CFREB)

Re: Certification of Institutional Ethics Review: Tool Support for Agile Iteration Planning

The above named research protocol has been granted ethical approval by the Conjoint Faculties Research Ethics Board for the University of Calgary.

Enclosed are the original, and one copy, of a signed Certification of Institutional Ethics Review. Please make note of the conditions stated on the Certification. A copy has been sent to your supervisor as well as to the Chair of your Department/Faculty Research Ethics Committee. In the event the research is funded, you should notify the sponsor of the research and provide them with a copy for their records. The Conjoint Faculties Research Ethics Board will retain a copy of the clearance on your file.

Please note, an annual/progress/final report must be filed with the CFREB twelve months from the date on your ethics clearance. A form for this purpose has been created, and may be found on the "Ethics" website, http://www.ucalgary.ca/UofC/research/html/ethics/reports.html

In closing let me take this opportunity to wish you the best of luck in your research endeavor.

Sincerely,

[Signature]

Patricia Evans
Executive Secretary for:
Janice Dickin, Ph.D., LLB., Faculty of Communication and Culture and
Chair, Conjoint Faculties Research Ethics Board

Enclosures(2)
cc: Chair, Department/Faculty Research Ethics Committee
Supervisor: Frank Maurer
CERTIFICATION OF INSTITUTIONAL ETHICS REVIEW

This is to certify that the Conjoint Faculties Research Ethics Board at the University of Calgary has examined the following research proposal and found the proposed research involving human subjects to be in accordance with University of Calgary Guidelines and the Tri-Council Policy Statement on "Ethical Conduct in Research Using Human Subjects". This form and accompanying letter constitute the Certification of Institutional Ethics Review.

File no: 4386
Applicant(s): Lawrence Liu
Department: Computer Science
Project Title: Tool Support for Agile Iteration Planning
Sponsor (if applicable): 

Restrictions:

This Certification is subject to the following conditions:

1. Approval is granted only for the project and purposes described in the application.
2. Any modifications to the authorized protocol must be submitted to the Chair, Conjoint Faculties Research Ethics Board for approval.
3. A progress report must be submitted 12 months from the date of this Certification, and should provide the expected completion date for the project.
4. Written notification must be sent to the Board when the project is complete or terminated.

Janice Dickin, Ph.D, LL.B,
Chair
Conjoint Faculties Research Ethics Board

Distribution: (1) Applicant, (2) Supervisor (if applicable), (3) Chair, Department/Faculty Research Ethics Committee, (4) Sponsor, (5) Conjoint Faculties Research Ethics Board (6) Research Services.
APPENDIX C: ETHICS APPLICATION

D.1 Consent Form

Name of Researcher, Faculty, Department, Telephone & Email:

Lawrence Liu, Faculty of Sience, Computer Science Department, (403) 607-8834, vpliu@cpsc.ucalgary.ca

Supervisor:

Frank Maurer, Computer Science Department

Title of Project:

Tool Support for Agile Iteration Planning

Purpose of Study:

My study is to investigate the software (AgilePlanner) that we developed, with regard to its effectiveness to the support face-to-face agile iteration planning processes. Agile methods are software development methodologies which are based on face-to-face interactions and colocated collaborations. The methods break the development life cycles into multiple iterations. Planning is held at the beginning of each iteration to discuss what needs to be implemented for the current iteration. At present agile teams usually adopt traditional planning media; for example, paper index cards, sticky notes, whiteboards etc, or traditional planning software; for example, MASE, Release Planner, Version One etc, to assist iteration planning. However, both approaches have drawbacks. At the front of traditional media information needs to be converted back and forth between planning and persisting activities. On the other hand, traditional planning offers limited support to synchronous face-to-face interactions; which prevents smooth team collaboration.

AgilePlanner addresses this issue by integrating technologies that are designed to allow the team members to engage iteration planning naturally with added benefits that computers can provide; for example, instant persistence and remote information access. You are chosen to participate in this case study for the following reasons:

1. You have been working on an industrial or course development project with the combined practice of XP and Scrum.
2. You know how iteration planning is handled in such a practice.

These background factors should enable you to interact with AgilePlanner with little effort. You will be observed or interviewed by the researchers, recorded by digital devices and surveyed with questionnaires. The gathered information will be synthesized into realizing the effectiveness of AgilePlanner.

What Will I BE Asked To Do?

The case study will monitor four successive iteration planning sessions. The duration of each iteration and the time gaps in-between each iteration will be subject to your development process. Your participation is voluntary, and you may refuse to participate altogether, may refuse to participate in parts of the study, or may withdraw from the study at any time without any penalty. There are altogether 5 questionnaires to be filled out and xxxx interviews to attend.
Questionnaire number one is to capture your satisfaction toward the use of the current planning tool. The feedback is used to identify elements that are beneficial or adverse to the current planning process.

Right after our tool suite is introduced to you, questionnaire number two is sent out to capture your first impression about the tool suite.

The tool suite will be used for two consecutive iterations, both of which will be videotaped for further analysis. After the first session of planning with our tool suite, the third sheet of questionnaire will be sent out for your feedback after the first use of the tool. The forth sheet of questionnaire gathers feedback after the second use of our tool suite.

In the very next iteration planning, you will switch back to old planning medium. The fifth sheet of questionnaire will be surveyed in the end of the planning to capture their opinions on the two different planning supports.

**What Type of Personal Information Will Be Collected?**

*No personal identifying information will be collected in this study, and all participants shall remain anonymous.*

There are several options for you to consider if you decide to take part in this research. You can choose all, some or none of them. Please put a check mark on the corresponding line(s) that grants me your permission to:

I grant permission to be audio taped: ____________________________ Yes: ___ No: ___

I grant permission to be videotaped: ____________________________ Yes: ___ No: ___

I grant permission to have my company’s name used: (only applicable to industrial participants) ____________________________ Yes: ___ No: ___

I wish to remain anonymous: ____________________________ Yes: ___ No: ___

I wish to remain anonymous, but you may refer to me by a pseudonym: ____________________________ Yes: ___ No: ___

The pseudonym I choose for myself is: ____________________________

You may quote me and use my name: ____________________________ Yes: ___ No: ___

**Are there Risks or Benefits if I Participate?**

*There is no obvious harm, discomfort, or risk associated with your participation in this research.*

**What Happens to the Information Provided?**

“Participation is completely voluntary, anonymous and confidential. You are free to discontinue participation at any time during the study. No one except the researcher and his supervisor will be allowed to see or hear any of the answers to the questionnaire or the interview tape. Any proprietary or personal information disclosed will remain confidential. There are no names on the questionnaire. Only group information will be summarized for any presentation or publication of results. *The questionnaires are only accessible by the researcher and his supervisor. The anonymous data will be stored for three years on a computer disk.*”
Signatures (written consent)

Your signature on this form indicates that you 1) understand to your satisfaction the information provided to you about your participation in this research project, and 2) agree to participate as a research subject.

In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. You are free to withdraw from this research project at any time. You should feel free to ask for clarification or new information throughout your participation.

Participant’s Name: ___________________________________________________________ (please print)

Participant’s Signature: __________________________________________________ Date:

Researcher’s Name: __________________________________________________________ (please print)

Researcher’s Signature: __________________________________________________ Date:
D.2 Case Study Design and Questionnaires

Case Study
Object of study:
The object studied is AgilePlanner’s tool interface.

Purpose:
The purpose is to evaluate the impact of the tool interface on agile planning meetings.

Perspective:
The experiment results will be interpreted from the viewpoint of the planning practitioners.

Context:
Testing subjects will be members of project teams that are running XP + Scrum development process. Project teams can be from industrial or academic environment. The team size and iteration length may vary according to the project team. Due to the limited number of Tablet PCs available, I propose to put a number cap of 8 as the maximum size of team that can participate in the experimentation.

Assumptions made by the researchers:
1. All members of a team are aware of their share of duties conforming to the XP + Scrum development process.
2. Before the introduction of AgilePlanner they perform iteration planning with a certain tradition planning tool, i.e. MASE or with index cards.

Content:
The case study lasts for 4 iterations. The four iterations are named I-a, I-b, I-c, I-d for illustration purpose. Tasks performed in this experimentation are broken down by iteration as follow:

I-a:
Questionnaire 1 will be given out in the end of this iteration planning. The purpose of the questionnaire is to find out how planning practitioners think about index cards or MASE according to their working experiences with it.

I-b:
The introduction of AgilePlanner will start with a tool tutorial which allows team members to learn how to plan with AgilePlanner. In the end of tutorial session, Questionnaire 2 will be given out. The purpose of the questionnaire is to collect users’ expectation toward AgilePlanner with reference to the past working experiences with MASE or index cards. Then the real planning session proceeds. The teams will not be told how long they should plan with AgilePlanner; however, they will plan with AgilePlanner for 2 iterations. In the end of planning sessions, Questionnaire 3 will be given out to. The purpose of the questionnaire is to find out if AgilePlanner lives up to the expectations of team members.

I-c:
Questionnaire 4 will be given out in the end of this iteration planning. The purpose of the questionnaire is to find out what tool operations with AgilePlanner are at odds with planning practitioners.

I-d:
Questionnaire 5 will be given out in the end of this iteration planning. During this planning session, the team members will be asked to switch back to MASE planning. The purpose of the questionnaire is to find out if they find AgilePlanner easier to plan with compared with MASE

Questionnaire 1:
1. What is your current planning media:
   1.1. Paper index cards (please answer question 2 - 5)
   1.2. Iteration planning software (please answer question 6 - 9)
   1.3. Hybrid of paper index cards and iteration planning software (please answer question 3,4,7,8,10 and 11)
   1.4. Others. (please answer question 12 - 14)

2. How do you store the iteration planning results?
3. Are there needs for remote access to iteration planning results in your team?
   3.1. Y.
   3.1.1. How virtual planning is achieved in your planning setting?
   3.2. N.
   3.2.1. Can you imagine any situations where virtual planning might be needed in your team?

4. What are advantages and disadvantages of using paper index cards for iteration planning?
   4.1. Advantages. (Please specify)
   4.2. Disadvantages. (Please specify)

5. Do you want to switch to computer supported planning tool?
   5.1. Y. Because of the disadvantages of using pure paper index cards planning you specify above?
   5.1.1. Y.
   5.1.2. N. What are other reasons?
   5.2. N. Because of the advantages of using pure paper index cards planning you specify above?
   5.2.1. Y.
   5.2.2. N. What are other reasons?

6. Have you encountered any difficulties clearly referencing a displayed task using iteration planning software?
   6.1. Y. An example?
   6.2. N.

7. Any tool operations you usually need to recall how you performed them last time?
   7.1. Y. Examples?
   7.2. N.

8. What are advantages and disadvantages of using iteration planning software for iteration planning?
   8.1. Advantages. (Please specify)
   8.2. Disadvantages. (Please specify)

9. Do you want to switch to traditional planning media (pen and paper index cards)?
   9.1. Y. Because of the disadvantages of using pure iteration planning software you specify above?
   9.1.1. Y.
   9.1.2. N. What are other reasons?
   9.2. N. Because of the advantages of using pure iteration planning software you specify above?
   9.2.1. Y.
   9.2.2. N. What are other reasons?

10. Why are you using a mixture of physical and virtual planning media?

11. How are you using these two types of media?

12. Please describe the planning media you use during iteration planning sessions?

13. Why does your team choose it as your planning media?

14. What are the disadvantages you see with the planning media?

Questionnaire 2:
1. After the tutorial for AgilePlanner, do you think AgilePlanner can replace your current planning media?
   1.1. Y. Because?
   1.2. N. Because?
2. What do you like or dislike about the AgilePlanner?
3. Do you think you can operate AgilePlanner without difficulties after the tutorial?
   3.1. Y. Why?
   3.2. N. Why?

Questionnaire 3:
1. Do you find the effect of AgilePlanner as what you thought it would be at the beginning of iteration planning?
   1.1. Y. Why?
   1.2. N. Why?
2. Do you think AgilePlanner can be used routinely in normal iteration planning sessions?
   2.1. Y. Why?
   2.2. N. Why?
3. Do you find any tool operations which you need assistance from researchers?
   3.1. Y. What are those tool operations?
   3.2. N. Roughly how many different types of planning activities you conduct today?
4. Do you find your collaboration with other team members differ from your old planning media?
   4.1. Y. Does it get better or worse?
      4.1.1. Better. How?
      4.1.2. Worse. How?
   4.2. N. I feel about the same.

Questionnaire 4:
1. Do you find the AgilePlanner become easier to plan with at the second time?
   1.1. Y. For example?
   1.2. N. I feel about the same.
2. Do you think AgilePlanner can be used routinely in normal iteration planning sessions?
   2.1. Y. Why?
   2.2. N. Why?
3. Do you find any tool operations which you need assistance from researchers?
   3.1. Y. What are those tool operations?
   3.2. N. Roughly how many different types of planning activities you conduct today?
4. Do you find your collaboration with other team members differ from your old planning media?
   4.1. Y. Does it get better or worse?
      4.1.1. Better. How?
      4.1.2. Worse. How?
   4.2. N. I feel about the same.

Questionnaire 5:
1. Compare your previous planning media with AgilePlanner, which one do you think is better for iteration planning sessions?
   1.1. Previous planning media. Why?
   1.2. AgilePlanner. Why?

References