Supporting Agile Project Planning

Lawrence Liu & Frank Maurer
Department of Computer Science
University of Calgary
{ypliu, maurer}@cpsc.ucalgary.ca

ABSTRACT

In this paper, we describe a tool supporting agile iteration planning. Our tool focuses on the support of human collaboration, which in turn allows the tool to be an integral resource within the planning process. Such support is enabled by utilizing advanced technologies, and accommodating the different types of planning environments, such as horizontal and vertical surfaces. The aforementioned integration not only distinguishes our application from traditional ones. More importantly it enhances the quality of the planning results. In addition to the said feature, our application can also be used to store planning data and monitor the progress of the project.

1. INTRODUCTION

Planning in agile software processes is done in meetings attended by all team members. Teams use paper-based index cards and the surface of a table or a whiteboard to define tasks, cooperatively estimate task effort and assign tasks to specific development iterations. While planning tools for agile teams exist, they are often ignored in the planning meeting. After the "real" planning concludes, one team member then enters the results of the meeting into the planning tool that then is used for progress tracking and reporting. We wonder what it is that stops the team members from using tools during the planning meeting?

It is our belief that current agile planning tools are too invasive and too slow compared to paper-based index cards. Our research question is: can we develop a collaborative planning tool for agile teams that combines the advantages of electronic media with the ease-of-use of index cards? To tackle this question, we first need to start with an understanding of the nature of agile methods and the iteration planning process.

Agile methods are being discussed and accepted by more and more software development practitioners. The core values of the agile methods are depicted by the Agile Manifesto [1] which has a dramatic shift of focus from Tayloristic methods [2]. Agile methods consider most software development knowledge as tacit and it is possessed by experts whose background falls in varying application domains. To tap into expert's in-memory knowledge, face-to-face communication and close team collaboration are preferred over heavy-weight documentation as a means of knowledge diffusion. The preference is also supported by a knowledge learning model proposed by Nonaka and Takeuchi [4].

One common activity in all Agile methods is iteration planning. Planning sessions provide learning opportunities for both business people and development team members. The quality of planning is largely subject to the quality of the information exchanged among team members, and the planning results are often preserved for future references and progress tracking.

While traditional planning software, i.e. MASE [6], enhances the use of planning knowledge, it does little to assist planning interactions by humans. Our observation is that planning tools are considered intrusive by many experienced agile practitioners due to the missing assistances. We assume the support provided by conventional planning software reduces the quality of knowledge exchange and capture, since less knowledge can be captured by tools if the exchange is challenged. Also, traditional planning tools usually require user trainings to master, while we believe iteration planning tools should include interfaces that simulate real agile planning environments. Such design should minimize knowledge loss induced by the intrusiveness of the tool operations.

Recent advances of various technologies now allow us to develop a solution to the aforementioned problems. Our research prototype combines supports from pen-input and mobility technology (Tablet PC), an amplified set of mice-based human computer interactions, and alternative projection technology (horizontal table display) with touch-sensitivity capability. Our purpose is to address issues commonly found in many conventional planning tools.

The pen-input system allows planers to interact with the software by a natural human gesture: handwriting. This support is in line with the use of index cards when agile teams are discussing the next iteration. The horizontal table display and newly developed interaction algorithms provide a virtual planning reality which requires a shorter learning time than that of traditional tools. With our tool, stories are created, updated and relocated in a similar fashion as paper-based index cards. This feature is implemented to avoid knowledge loss that results from the separation of data generation and data recording. We also implemented support for mixed public and private workspaces. This mixture empowers distributed planning and allows team members to release their changes when they choose to. After a change is released, all other registered planners will be informed with an update to their own workspaces.

In Section 2, the XP planning process is described and problems observed in real planning sessions with regard to current tool usage are presented. Technologies employed by our research prototype are discussed in Section 3. A scenario example is also included in the same section to illustrate how the tool can be used in a planning setting. Our future research is outlined in Section 4 which articulates the importance of conducting an empirical study and collecting metrics for this research. Finally, we conclude by

highlighting how our tool would help a real planning game and potential issues.

2. The XP Planning Game

Extreme Programming (XP) encloses planning activities in the Planning Game. The planning process starts by creating "stories" (the XP term for high-level user requirements), estimating the development effort for all stories, calculating the overall effort for an upcoming iteration and comparing it with the available effort, prioritizing stories and deciding on a story implementation sequence. Stories serve as task reminders to developers. They contain enough story information that developers can feel confident to make estimates on the tasks. Implementation details can be addressed by further customer-developer collaborations at later times. Without software tool support, stories are usually written on a paper index card. Paper index cards are chosen as a physical representation of a story for their ease of use when people collaborate. Additionally limited space of paper index cards helps narrow story writers' focus down to the high level abstraction of the task. Stories can be assigned to a selected iteration or the product backlog. Those in iteration are scheduled to be completed by the iteration's completion date, whereas ones in the product backlog will remain undecided for implementation until their time has come. Completed stories are erased from the memory of the project, whereas unfinished or immature stories are preserved for future reviews. Dispatching human labors to safeguard, maintain and indexing piles of story cards is uneconomical.

The effect of the planning game should be enhanced by the use of electronic tool support. Computation technology enables persisting stories in a central repository, public story access for valid personnel, and instant indices to stories of request. However, planning software should be more than a decision storing and tracking system. It should be an integral part of the planning meeting instead of being filled with the information afterwards. Our research wants to find out obstacles that planning members have with traditional planning tools, and validate the effectiveness of our tool by examining how many the discovered impediments can be lifted to what extent.

3. TOOL SUPPORT

3.1 Technologies Used

Tablet PCs are equipped with stylus-sensitive screens. Pen motion is tracked and buffered for handwriting recognition through a software recognizer. The accuracy of handwriting recognition technology deployed on Tablet PCs is observed to be reasonable. Higher accuracy is expected to be reached as many commercial venders are working on this area. The PC interacts with the pen the same way it does with mouse. Tablet PCs are designed to be used like real notebooks. They are lighter and easier to cope with even in positions that are considered awkward by the use of traditional laptops. For example, it is really hard to stand and type at the same time with conventional laptops, when it can be easily done with Tablet PCs. The same awkwardness can be experienced when exchanging ideas. It always distracts the person that is holding a laptop from truly listening the other person's speaking. With a Tablet PC, a person can easily secure it in one arm without constantly worrying about dropping it. Because of the design parameters and intuitive input system, we believe that a Tablet PC is a better communication tool than traditional laptops.

A horizontal display with touch-sensitivity capability is the foundation of our planning workspace. Just like planning with paper index cards, image objects on the table display can be piled in any manner as planners' wish. Also since agile iteration planning is usually carried out on a table, a horizontal display is used to simulate the planning environment.

Our tool is based on new interaction techniques developed by Dr Carpendale's group at the University of Calgary for the horizontal display. Objects on the screen can be dragged by the corners and they will rotate instead of plainly moving horizontally or vertically [9]. The object can also be thrown from one any location on the table to another. Throwing is usually a better solution when object is passed a distance too far for dragging. The visual effects of rotating and throwing create a more "real" virtual planning environment. For instance, throwing an image objects to a particular person sitting far away at the horizontal display seems to be a more "natural" interaction than using a mouse to move an object across the table. Also planners can always rotate the display of an image object to suite their own visual orientations while reading the card (this is not an issue for vertical displays in agile planning).

3.2 Example Scenario

Imagine the following scenario: A team of 8 people use agile practices for developing a software application. The team includes both business and development experts. They use our tool for supporting their planning meeting. Keep in mind that this electrical planning can be done on a horizontal table display as well as a vertical display. Figure 0 and Figure 1 demonstrate the planning tool with these two kinds of display.

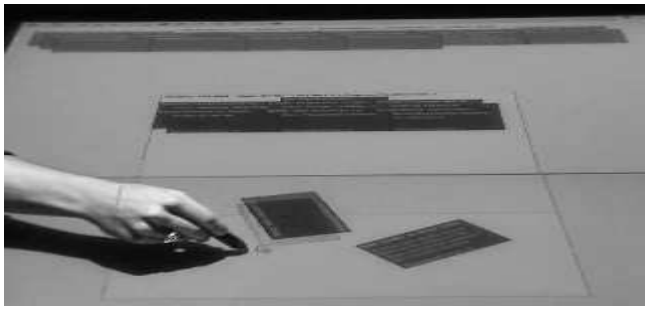

Figure 0: Planning with pen and a table display our tool.

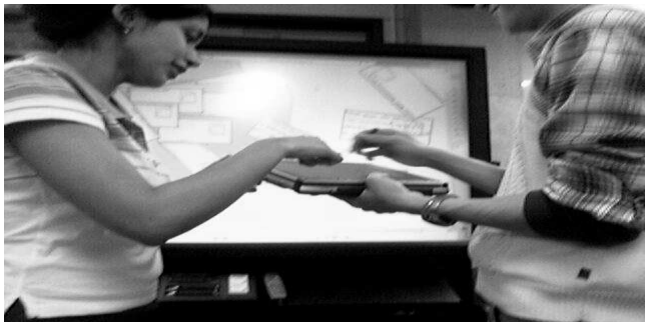

Figure 1: Collaborating with pen and a table display with our tool.

At the beginning of the planning the team decides the deadline of the iteration and an empty iteration is created. Iterations are graphically represented as a green rectangle area in the application. Another purposeful section on the screen is story completion area. Stories that are placed into this blue area are considered completed for the selected project. The rest of the screen space belongs to the product backlog. Every agile project has a product backlog that is used as a wishlist for additional features or stories. Iteration and story completion area are floating objects which can be moved within the displaying area. Stories can be placed arbitrarily on the screen like planning with paper index cards on a real table. Story and iteration locations are "memorized" in a central database. The resulting visual effect is equivalent to taking a snapshot of the planning result. Upon revisit it helps recap planners' memory regarding the activities that occurred in the previous iteration planning. Each story card is represented as a colored rectangle. There are three colors each of which represents the task nature of a story. Khaki is for bug fixing; Honeydew is for new feature; and Light sky blue is for refactoring. Story cards can be displayed with any rotation angle which is useful for developers who sit at different sides of the table display. Figure 2 shows an initial state of a planning session. Stories and iteration can be "thrown" around, which is also something people do with paper index cards in agile planning meetings.

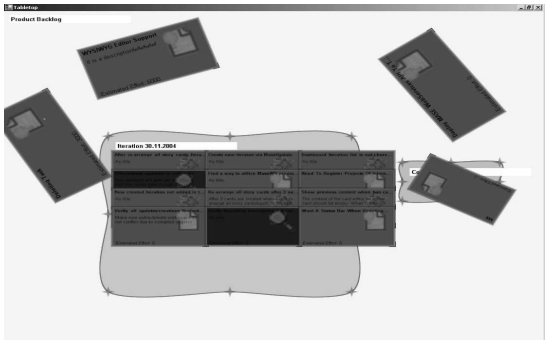

Figure 2: Planning session

Creating a story is also an easy operation with this tool. A story editor can be brought up by clicking on the story creation items on the menu. This can either be done via a mouse click or the planner's finger. On a Tablet PC, story content can be entered by either typing or writing. Information supplied are story name, responsible & pair programmer, the task nature of the story, effort estimations, and story description. The form is identical as the example index card in K. Beck's book on Extreme Programming [8]. With handwriting, story content can be modified with pen and eraser. Figure 3 shows a handwritten electronic index card.

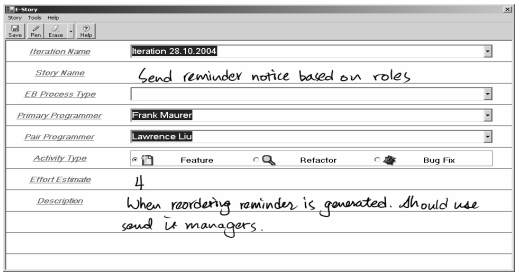

Figure 3: Handwritten electronic index card

Once the handwritten story is saved, it is displayed in the planning workspace. Figure 4 demonstrates how a handwritten card will look in the planning workspace. The new story is streamed up to the MASE [6] system, an online e-planning tool also developed by EBE [7] research group in UofC, so that planning results can be shared and accessed within a distributed team.

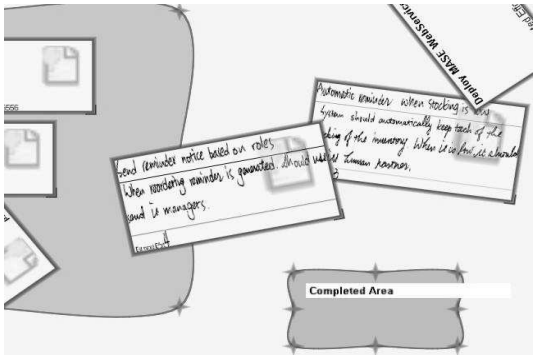

Figure 4: Handwritten index card in the planning workspace

Distinguishing between public and private workspaces is another feature of our tool. When using the tool in private workspace mode, no planning change is released onto the MASE server and, thus, it does not become available in a shared/public workspace. As a result, the public planning workspace can be free from frequent data updates unless a team member really wants to share new information with the rest of the team. Changes released by a public workspace can be received by workspaces of either mode. This update mechanism provides a means for the distributed workspaces to synchronously cope with new releases.

4. FUTURE WORK

The tool implementation is now completed but our original goal of developing a less-invasive planning support tool is not yet empirically validated. To investigate the effect of the tool with regard to iteration planning, we will collect data with reference to the following questions.

1. Ease of use of the tool. Invasiveness is what our assumptions base on to explain why tradition planning tools are not used primarily for collaborations. Is the new tool really less invasive? By our observation when people are introduced to this new tool the answer seems to be promising. Then, by how much more are people willing to use our planning tool over traditional ones? What are the reasons for them to use our tool and whether or not rationale identified is inline with our assumptions?

2. The effect of intuitive interface with reference to social culture building among team members. Team building heavily bases on human interactions. Since the tool supposedly free planners from distractions induced by traditional planning tools, do planners feel that they can better explain themselves with the new tool? How close they feel the new tool operations is to the natural human interaction process? Does that intrusiveness gain any edge for teams to build an engaging planning culture?

Experiments will be conducted with XP teams consisting of university students as early as January 2005.

5. CONCLUSION

Planning support tools should be more than just result entry systems. We propose that agile iteration planning can be better served with a tool that supports a natural means of human communications, i.e. writing, sketching and "throwing" image objects on a table like paper index cards. Also keep in mind that agile practices place heavy focus on human interactions, which further fosters the demand of an intuitive planning tool during agile planning. Our tool is believed to encourage planning participants to collaborate in manners more streamlined with their natural expressions. Elaborating tacit knowledge by itself is difficult, and it is more so when the planning tool is distracting. Counter arguments can be our tool's heavy focus on the intuitiveness is overrated because people have ability to adapt. It remains unclear if planners are forced to collaborate with traditional planning tools for a period of time could they find improvements in using our implemented tool. It is observed that many experienced agile planners expressed personal preference in our tool which we consider positive feedback. In the near future, we are going to apply this planning prototype in controlled experiments and will compare it with current agile planning tools as well as with paper-based agile planning processes.

6. REFERENCES

[1] http://agilemanifesto.org/

[2] T. Chau, F. Maurer, G. Melnik: Knowledge Sharing: Agile Methods vs. Tayloristic Methods, Proc. WETICE 2003, IEEE 2003.

[3] Technology Support for Social Networks in the context of Task-based Knowledge Management, Ryan Fernendes, Henry Linger & Frada Burstein

[4] I. Nonaka, "A Dynamic Theory of Organizational Knowledge Creation", Organization Science, Vol. 5, No. 1,1994, pp. 14-37.

[5] http://ebe.cpsc.ucalgary.ca/Frank.Maurer/assets/mase.wmv

[6] http://mase.cpsc.ucalgary.ca/ebe/

[7] http://ebe.cpsc.ucalgary.ca/Frank.Maurer/

[8] Kent Beck, "Extreme Programming Explained: Embrace Change", Addison-Wesley Pub Co; 1st edition (October 5, 1999)

[9] Kruger, R., Carpendale, S., Scott, S. and Greenberg, S. (2003). How people use orientation on tables: Comprehension, coordination and communication. *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (Group) '03*, ACM Press, pp. 369-378.

7. APPENDIX

The demonstration will mainly focus on the intuitiveness of our planning interface. Dragging, throwing rotating an image story card will be illustrated like the movie clip, http://ebe.cpsc.ucalgary.ca/Frank.Maurer/assets/mase.wmv. We will also show how other planning requests can be satisfied by using our tool. To start, the team needs to authenticate themselves as valid users to the system. It is done through a login interface shown in Figure 5.

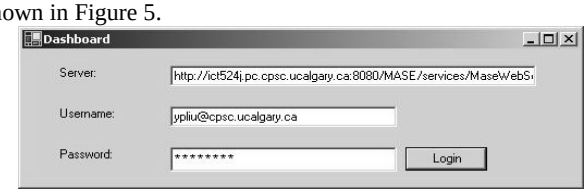

Figure 5: Logging into the system

The team then needs to specify the iteration that will be planned for. The iteration selection needs the following information: the project that the iteration is under, the name of the iteration, and the planning modes. The difference between the two planning modes, public and private workspace, will also be discussed. Figure 6 shows the window to provide information needed for the iteration selection.

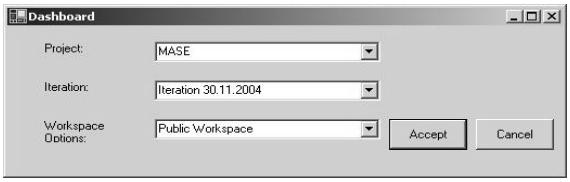

Figure 6: Selecting an iteration that will be planned for

To plan for a new iteration, team members click on "Create New Iteration" button available through the administrative menu that is presented as Figure 7. The administrative menu also provides access to many other administrative tool operations.

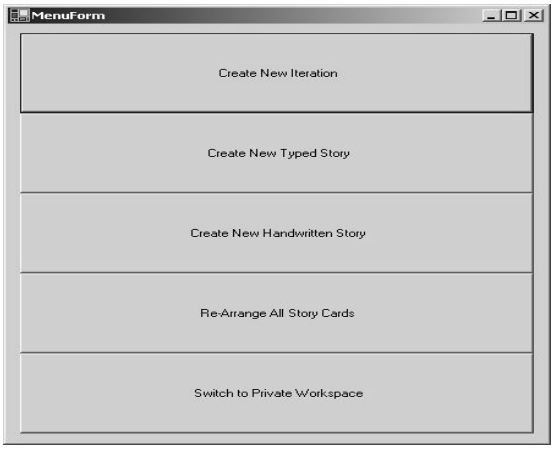

Figure 7: Administrative Menu

When a new iteration is created, iteration name and completion date are needed. The completion date can be specified by circling the expected date from the expended calendar. Team will use the following interface, Figure 8, to create a new iteration.

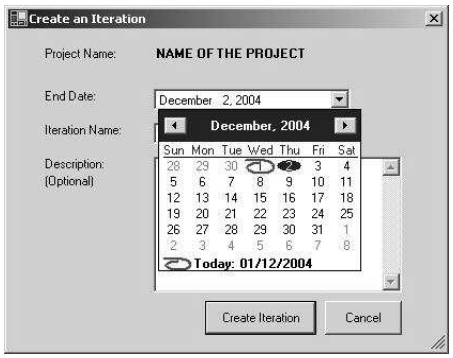

Figure 8: Creating a new iteration

Initially the created iteration contains no stories to be fulfilled. Team can drag or throw stories from the product backlog of the project to the iteration. The result of multiple times of dragging and throwing will look like Figure 2. Stories inside the green rectangle area belong to the selected iteration, while others are part of the product backlog. New stories can also be included into the planning by clicking on "Create New Handwritten Story" or "Create New Typed Story" buttons available on the administrative menu shown in Figure 7. Creating new handwritten story will be done with a Tablet PC. Easy human collaborations over the use the pen input system will be illustrated. The demonstration will need four volunteers from the audience. Two of them will be using MASE to create a story, when the other two are using our application. The story to be created will contain tacit knowledge that requires sharing from one end to the other. Willingness of using the two tools will be captured by asking the two pairs after the story is created.

Once the story content is finalized, the story can be saved onto the MASE application server. The story creation interface is displayed by Figure 3. If the planning is done in the public workspace mode, the story created will be stored onto the server database and other planning clients using the same application tools will be informed with an update notice. Such notice will trigger a refresh of the planning surface, and newly created stories will be displayed in all planning client's workspace. If the planning is done in the private workspace mode, only the local planning surface will contain the updates. If the team wishes to release their changes to the other planning sessions, they can click on "Switch to Public Workspace" button available on the administrative menu. The planning mode can be switched at any time during the planning.