VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

e-Wall: a Multi-Touch Agile Story Wall

Matthew Crisp

Supervisors: Dr. Rashina Hoda
Dr. Stuart Marshall

Submitted in partial fulfilment of the requirements for
Bachelor of Engineering.

Abstract

Story walls are an integral part of Agile software development and are typically a paper artefact. Electronic versions of story walls exist to provide benefits over their paper based counter parts, such as distributed collaboration, content generation, or content archival, but take away from the traditional form of interaction a paper wall has. This paper describes the creation of an electronic story wall focusing on dynamic space utilisation which was implemented on a multi-touch table to maintain the original feel and interaction of a paper wall. This prototype was evaluated by current Agile practitioners, and international Agile researchers.

Acknowledgments

Thanks to my supervisors, Rashina Hoda and Stuart Marshall, for their input and guidance throughout this project. Thanks to the HCI group for support, in particular Craig Anslow for building the multi-touch table and assisting in its use. Thanks to the local and international experts who participated in the evaluation and provided valuable feedback on the prototype.

Contents

Figures

Chapter 1

Introduction

A key component to agile software development is the story wall. It is what development teams use to manage and track their progress of work throughout an iterative process. Traditionally, story walls are physical artifacts that have cards stuck on them containing descriptions of work chunks that make up a subset of the project. The wall becomes a central area that all team members must stand in front of when consulting for a new item of work to do, or updating the status of one already being worked on. This creates situational awareness[9] of the project as the whole team can see who is making changes, and what those changes are.

Paper walls gain no benefit from the technological world we live in. The wall is static, any changes to the wall must be made by hand and can be time consuming. Cards on the wall are prone to damage and can easily be lost. Sharing of information between distributed teams requires the duplication and maintenance of the walls information in electronic copies.

Electronic walls offer solutions to some of the downfalls of paper, but suffer from their own limitations. Moving the wall into a digital medium takes away the traditional form of interaction and replaces it with a keyboard and mouse. The wall should be a central piece of focus for development activity that is easily visible to all developers, allowing them to walk up to it and interact with it. These interactions get left out when the wall is digitised onto a desktop. Each team member works with their own view of the board, there is no situational awareness around it.

This project involved creating a prototype of an electronic story wall to provide the benefit a dynamic layout and space utilisation. To maintain the classic interaction of a paper wall, this prototype was implemented on a multi-touch table.

1.1 Contributions

The contributions of this project are:

- A dynamic layout and interactive visualisation of a story wall.
- A proof of concept prototype implemented on a multi-touch table to preserve a more traditional form of interaction.
- An evaluation of the prototype by local domain experts and international researchers.

1.2 Report Structure

The rest of this report is structured as follows: Chapter 2 provides a background along with related work to do with agile story walls. It also provides a description of the methodology this project followed. Chapter 3 provides the scope of the project through an analysis to alternative solutions. Chapter 4 talks about the design of features throughout the iterations of the project. Chapter 5 describes how MT4J is used to create the prototype. Chapter 6 provides results of a user evaluation and heuristic evaluation on the prototype. Chapter 7 concludes with future work.

Chapter 2

Background & Related Work

In this chapter, section 2.1 describes what Agile development is, what a story wall is, and what a multi-touch table is. Section 2.2 gives an overview of related work on story walls and agile tools. The final section, 2.3, describes what process this project followed and gives a timeline.

2.1 Background

2.1.1 Story Wall

Agile development is a group of methodologies that take a lightweight approach to software development[2, 8]. The focus of Agile is to develop software using short iterative cycles, aiming to produce working code over comprehensive documentation. Agile promotes teamwork trough close proximity and intense interaction. Some popular Agile methods include Scrum, eXtreme Programming (XP), and Kanban. In Scrum, a 'sprint' is a time period when development on a set of items occurs. During a sprint, only stories that are part of the sprint appear on the story wall.

Planning iterations under an Agile methodology involves the creation of story cards. A user story captures a feature about a system that is under development, and the story is written on a piece of card. Stories are broken down into tasks that are required to be completed in order to have the feature that the story describes. To track the progress of these stories, a story wall is used.

The story wall is a central piece of focus where the story and task cards associated with the current iteration are kept. A story wall is shown in Figure 2.1. The wall is divided up into columns that represent the status of the cards and they are moved through the wall by team members to reflect the progress made on them in the system. The exact number and labelling of these columns differ team to team. The most basic form would be:

- To Do: holds tasks that have been planned for the iteration but have not yet been started
- In Progress: holds tasks that are currently in development
- Done: holds tasks that have been completed

Users of the Scrum methodology use a 'scrum board' which normally has the three columns described above. Kanban users tend to have more varying names and numbers of columns.

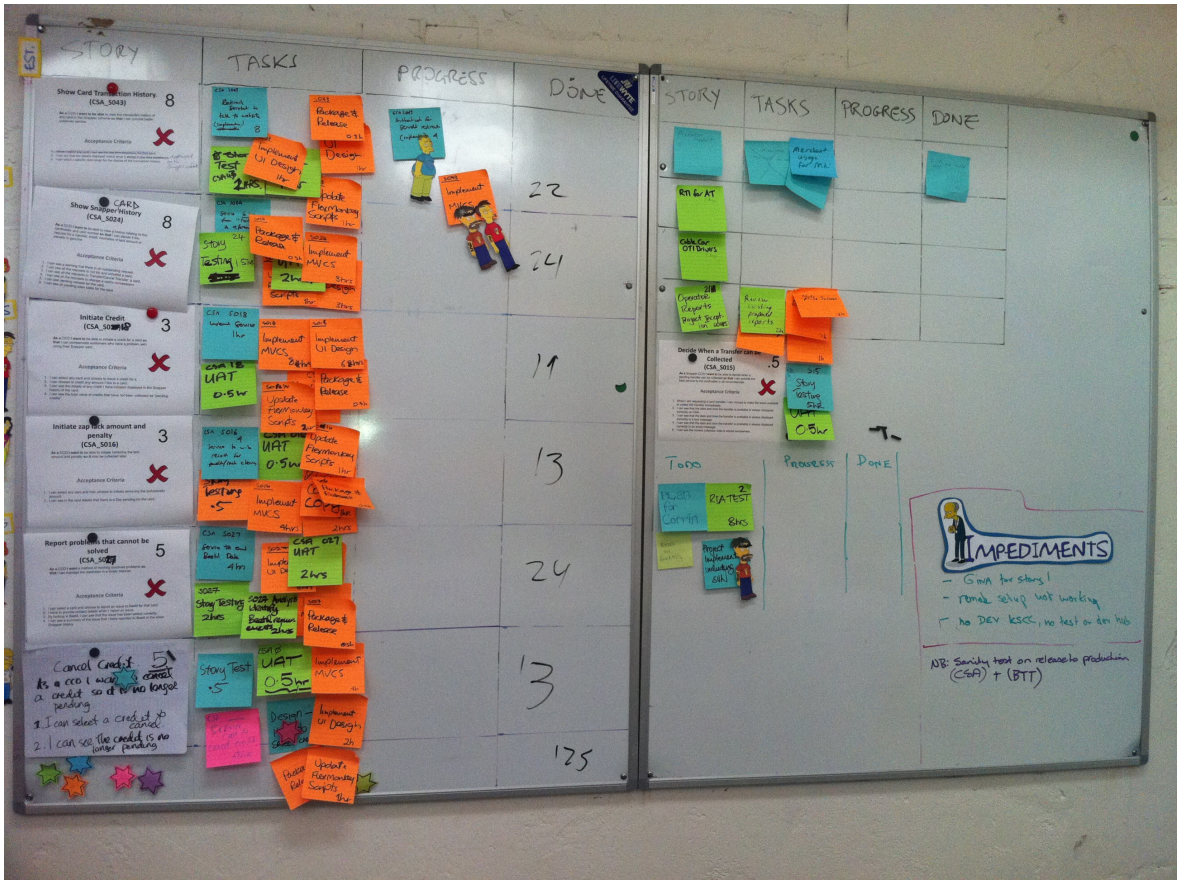

Figure 2.1: Paper story wall.

2.1.2 Multi-touch table

A touch table is a large horizontal surface that can be interacted with by touching it. A multi-touch table responds to two or more touch points, this allows for more complex gestures to be made on it when compared to single touch. The multi-touch table used in this project is shown in Figure 2.2. The table has two internal components, a camera for picking up touch points, and a rear projector for the display.

Mt4J[5] is an open source multi-touch framework that supports multiple input devices. It is the framework used to develop the prototype, and is compatible with the multi-touch table used.

2.2 Related Work

2.2.1 Story walls

The story wall provides a central location of information about the current iteration of a project[9]. Team members can walk up and interact with it, have discussions, and make decisions about what they want to do next. Cards on the wall are tangible artefacts, to interact with them they need to be picked up and physically handled. Having the wall in a visible location provides a high level of situational awareness [9]. Consulting the wall involves standing in front of it as team members have to physically handle cards . Other members can see that someone is in front of the wall and interacting with it. The changes the make are instantly visible and others are immediately notified of these changes due to

Figure 2.2: Multi-touch table.

its central location. This gives the team a feel for what each other is doing, and encourages communication between the them, a key principle of agile development[2].

The problem with having cards on paper is they gain no benefits from technology. Without being in a digital medium they are inherently static, offer no means for quick storage or look up of information, prone to damage, and difficult to share with distributed teams. Studies[8] into story cards have listed the disadvantages of paper cards as:

- No support for auto numbering
- No copy/paste
- Deletion/amendment difficult and messy.
- Single copy prevents sharing
- Possible loss of copy

In an attempt to address the above issues, a range of digital card systems have been created[1, 4, 6, 8, 10].

DotStories[8] is a user story tool that offers the creation and maintenance of stories. It contains stories in a three levelled organisation of websites. The top most level corresponds to projects, the next level down are a collection of story groups, and the final level contains the user story and related information. DotStories targets the list of problems above by providing the missing features of the paper cards. The problem is that DotStories only maintains a collection of stories, it does not provide a wall to track the stories during development. Duplication of artefacts would occur in a development team. They would first have the story in DotStories, and then again on a paper wall.

AgileZen[4] and LeankitKanban[1] are two tools that do feature a virtual wall. They allow for teams in distributed locations to access central information containing stories and virtual cards on a wall. Users can move these cards across the wall into the various stages using their mouse. These tools provide the features that paper cards are missing, and include a virtual wall to display and track their progress, but in solving the problems, they have removed what makes a paper wall effective.

Situational awareness is an important aspect of paper walls[9]. Having the wall contained in a computer removes this awareness. The wall isn't tangible in tools like AgileZen and LeankitKanban, team members do not have to walk up to it in order to interact with it. It is no longer a central part of the work space that encourages communication and team work. Sharp et el[9] states that the problem with electronic walls is that they obscure this information flow.

If agile tools are to be developed, research[3] has shown that they must fulfil the following requirements:

- Lightweight, only have necessary features.
- Easy to use
- Flexible, allow for customisation and configuration.
- Offer some benefit, such as automating boring tasks or supporting communication and cooperation
- Be accessible for everyone
- Make team members aware of any changes

2.2.2 Multi-Touch Agile Tools

MasePlanner[6] and APDT[10] are two tools that allow for planning of story cards to be carried out on a multi touch table. These projects are aimed at providing benefits of planning in a distributed environment whilst maintaining the feel of interaction that closely matches a physical medium. The difference between these and our project is that they do not have a story wall. MasePlanner and APDT are both planning tools. They are used during creation of the story cards and planning of iterations where as the story wall is used throughout the iteration as a tool that tracks and displays the progress of cards. Using a multi touch table as a story wall would allow for digital benefits to be in place, but retain the situational awareness and interaction of a physical artefact. There are currently no multi touch story wall solutions that retain advantages of a physical medium through a touch table whilst benefiting from features a digital tool can provide.

2.3 Methodology

We used an Agile approach in this project. This is an incremental approach featuring iterations. The goal at the end of an iteration is to produce a working prototype with a subset of the total features. The design and implementation went through three iterations. This served two purposes: it helped manage the various stages of the project in an effective manner, and helped us understand the domain better as we were building an Agile project management tool. Following this approach, each iteration contained frequent meetings with my supervisors to gather feedback and input and user stories were written to capture the features required by the end of the iteration. To help planning, a demonstration of each prototype was given to either domain experts or my supervisors at the end of each iteration. This feedback fed into the development of the application in during the next iteration.

The timeline for the iterations is given below:

- Iteration 1 [March 28th - May 2nd]
  Create a basic story wall prototype that replicates a paper based wall
- Iteration 2 [May 2nd - July 11th]
  Add benefits of dynamic space utilisation to initial prototype of story wall
- Iteration 3 [July 11th- September 5th]
  Finalise the prototype by updating the UI and implementing new features

The Gantt chart used throughout the project is given in figure 2.3

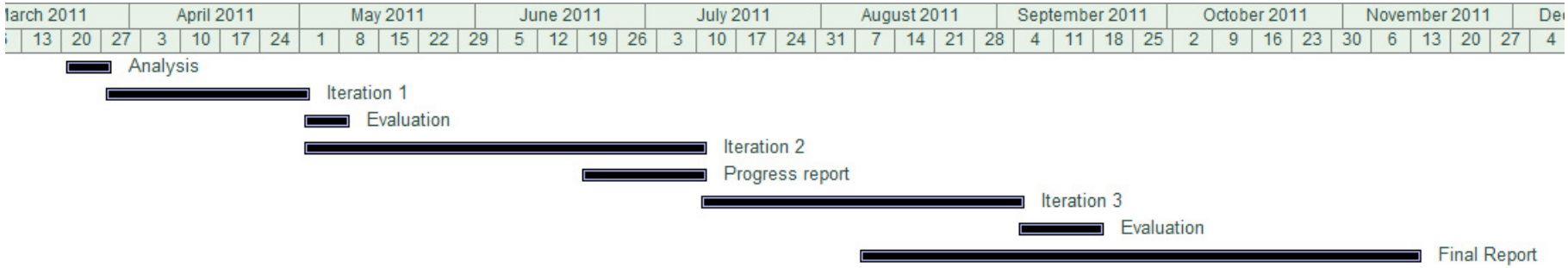

Figure 2.3: Gantt chart.

Chapter 3

Analysis

The initial idea for this project was to create an electronic story wall that would aim to solve a problem story walls have when in paper form, as described in section 2.3.1. This chapter describes the alternative approaches of implementing an electronic story wall that were considered.

3.1 Platform Alternatives

To create an electronic story wall, decisions around the scope of the project needed to be decided early on. These initial decisions involved what kind of story wall would be developed, and what kind of domain it would be developed for. Through literature review and discussion with supervisors, four major options were explored.

- Desktop application
- Web application
- IDE plug-in
- Multi-Touch application

To make a decision on what option to take, the pros and cons of each domain were weighed up. These are described in the following sections 3.1.1 through 3.1.4

3.1.1 Desktop Application

Developing a story wall as a desktop application would involve creating a standalone application for managing a project. This option would create the most basic form of a story wall. The application would provide a virtual wall that holds tasks and story cards. These cards can be moved along the wall by the user of the application in the traditional mouse and keyboard form of input.

The benefits of implementing a story wall as a desktop application are that certain aspects of the wall can be automated. Progress of the project can be automatically analysed using the data extracted from the wall. For example, charts can be created to show how much work has been completed, and how much more there is to do. This is something that would be time consuming if done by hand. Using a digital wall can save the time it takes to maintain artefacts related to the management of a project. The less time spent on creating documentation leaves more time for development, an important part of agile development.

The limitation of such an approach is that the wall is only accessible from a desktop computer. Team members have no central entity to consult about tasks. This results in a

reduction of situational awareness and wall is often duplicated in a paper and digital form. This duplication wastes time, but without it information flow is affected.

3.1.2 Web Application

Implementing a story wall as a web application would see a solution available from a web browser. A virtual wall would be created that could be accessed from anywhere with Internet. Like the desktop application, story and task cards could be created and moved through the virtual wall by the user. The difference from the previously proposed solution is that it would be built on web technologies and lie in the web domain instead of the desktop domain.

The main benefit of having the story wall as a web application over a desktop one is the ability to easily support distributed teams. This would provide a solution that could be accessed off site without the need for any additional software to be installed. Having the wall accessible from the web would help agile teams that work from multiple locations to stay informed and allow for greater collaboration.

A web based story wall would suffer from the same limitations as a desktop solution. The wall is not an entity in the workspace of team members. Developers sit at their computers individually to manage tasks. The web story wall does not provide the interaction developers get with a paper wall.

3.1.3 IDE Plug-in

Creating a story wall for integration into an IDE would provide a solution that gives developers instant access to the wall. Like the previous solutions, a virtual wall would be created, but this time it would reside directly in the environment the developer is working in. Having the wall in the development environment opens it up for possibilities to collaborate directly with source code. A task could be link directly to a specific section of code. The wall could also collaborate with the testing suite, providing information about what tasks are passing their tests.

3.1.4 Multi-Touch Application

Putting a story wall on a multi-touch table would open up a world of interaction not possible with previous solutions. It would preserve the feel of a paper based wall, allowing for collaboration from multiple team members at once. An interactive virtual wall would be created that could be interacted with through gestures on the table. Cards would be moved around the wall in a way that is most similar to how you would move a card on a paper wall.

This is a best of both worlds approach. Having the wall in a digital medium allows for the benefits of a dynamic wall that can automate tasks and generate content. It also maintains the feel and interaction of a paper wall, keeping it as an entity that multiple team members can walk up and interact with.

3.2 Chosen Solution: Multi-Touch Application

It was decided that the story wall will be developed as a multi-touch application. This decision is based on the interaction methods of all four choices. Desktop, Web, and an IDE plug-in all use the standard mouse and keyboard combination for manipulation. To retain the feel and presence of the agile story wall as discussed in section 2.2.1, it needs to be a

standalone entity that promotes interaction. Using a multi touch table allows for it to be a dedicated piece of hardware that team members can come up to and use without a keyboard and mouse getting in the way of the interaction.

There also exists other desktop[8] and web[4, 1] solutions, however, no multi-touch story wall solution exists.

3.2.1 Scope

To define the scope of the project, we decided to focus on a main area of benefit a digital medium can provide to emphasise in the story wall prototype. We came up with four possible areas we could focus on.

- Collaboration
- Content tracking
- Content generation
- Dynamic space utilisation

Collaboration would involve exploring ways that the multi-touch table can allow for multiple users of a story wall at the same time, both co-located and distributed.

Content tracking would look at implementing ways to track the movement of cards along the wall, and replaying it to show the walls history.

Content generation is closely linked with content tracking, and would see supporting artefacts being generated, such as burn down charts.

Space utilisation looks at how the computer can assist in the arrangement of the storyboard to best use up the space available whilst showing important information.

Common areas that all these options share are the investigation of interaction with a story wall on a multi-touch application, and to preserve the feeling of interaction with a paper wall.

Chosen Target Area: Dynamic Space Utilisation

The decision was made to narrow the scope of the project to concentrate efforts on way to most effectively use space available. This decision came about from the nature of the touch table. They are large tables but do not run in high resolutions such as 1920X1080. This means that the screen real estate is limited as to how much they can show at one time. Paper based story walls are also static, they cannot dynamically change and adapt to optimise space utilisation. This is the basis for the decision to investigate ways that the computer can most effectively use space displaying a story wall and what benefits these have over a standard paper based wall.

3.2.2 Prototype Components

A completed story wall application would see 3 components that make up the system.

- Product backlog
- Story wall
- Archive and metrics

The lifespan of a card in such a system would start at the product backlog. This is the planning and creation stage where brainstorming actions would take place. Data input would be needed to get information about tasks into the system, and manipulation actions should exist to allow the team to categorise and play with the tasks. Once story and task cards have been decided, they would be prioritised and added into the product backlog list.

Once the backlog is populated, stories can be added onto the wall component. They have each of their tasks moved along the various stages of the wall until all the tasks have been completed. Once this happens, the story card is then taken off the wall and moved into the archive.

The archive and metrics component would serve two functions, providing a means to examine previous cards associated with previous iterations, and generating supporting artefact's using information from the system.

Chosen Component: the Story Wall

The prototype being developed in this project is the wall component. The area of investigation is how the wall can be implemented on a multi-touch table, how people could manipulate it, and how it can effectively use its dynamic nature to assist in space utilisation.

The product backlog area has not been investigated for two reasons. Firstly, there are similar applications that aid in the planning of projects on a multi-touch table[6, 10], and secondly, the nature of data input on a multi-touch table could be a project in itself. It is important to provide an accomplishable scope for this project, and as such, resources should be directed into the most useful area.

The archive and metrics component has not been investigated because of the considerations in section 3.2.1. Archive and metrics encompasses two areas of investigate that were decided against, content tracking and content generation. Because the main focus of the project is to investigate how the wall can assist in space utilisation and manipulation, archive and metrics lie outside of the scope.

Chapter 4

Design

This chapter provides a narrative of the three iterations the project went through. It introduces the features added in each iteration, and the reasoning behind them. Design was not carried out before implementation, but alongside it because of the iterative approach taken. At the end of this chapter is a summary of the features included in the final prototype.

4.1 Iteration One

At the start of the iteration time was spent conducting a literature review of related work such as agile planning tools on multi touch tables and digital story walls. These are discussed in chapter 2.

The end goal of this iteration was to have a very basic working story wall that allowed cards to be moved around. In the early stages, possible angles such as collaboration, content generation, and archiving were all considered as target areas of the project. As the main goal of this iteration was to implement a standard story wall, it was important to get a working prototype that matched a paper based wall before we started to add benefits of a digital wall.

In order to start creating a multi-touch wall, a few weeks had to be spent learning the MT4J framework[5]. This consisted of small mock programs to familiarise myself. These mock programs consisted of small hierarchies of shapes that would later form the basis for the first story wall.

Once familiar enough with MT4J to start creating a simple story wall, the weekly meetings became a time to discuss and plan what tasks would be carried out over the coming week in order to have a simple story wall by the end of the iteration. The planning of these was conducted in an agile way with user stories being formed along with tasks to go with them. This allowed me to get hands on experience using my own paper based story wall while creating a digital version.

The prototype created by the end of iteration one was a simple grey wall with 3 columns representing the stages of; To-do, In progress, and Completed. Story cards were represented as a yellow rectangle. In this prototype tasks were not present. The main features of the prototype at the end of iteration one, shown in Figure 4.1, were:

- The ability to drag a story card using a drag gesture with one touch contact point. Dragging the card allows it to be moved along the storyboard. The card recognises when it is dropped into a new stage and snaps into place along the left margin of the column.

- The wall its self supports drag, scale, and rotate. This was an important feature as we were still open to the idea of collaborative use where the wall may have to be passed around and re orientated for different users around the table.

- A story card could also be rotated and scaled, allowing it to be moved around the table to different users, again, this was important as we were thinking about a collaborative angle to using the wall. Once a card is then dragged over the wall and released, it snaps into place, resetting the size and orientation to match that of the wall.

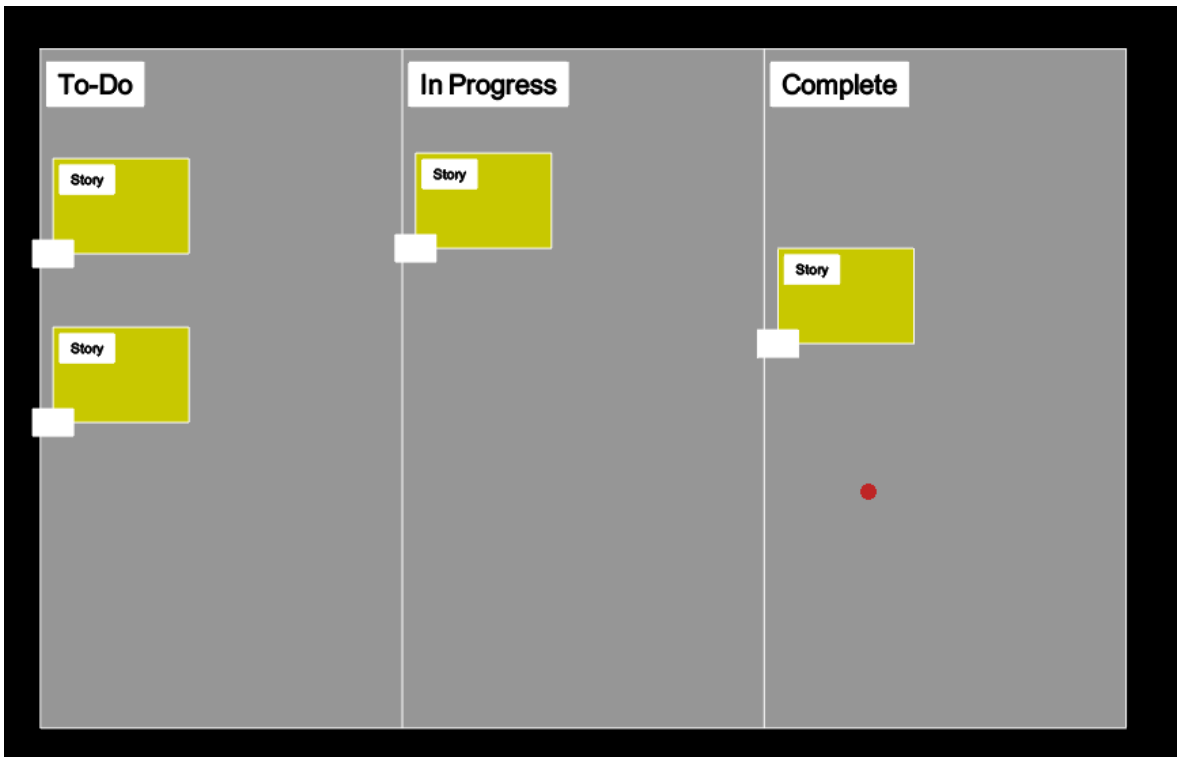

Figure 4.1: Initial prototype of story wall. Shows four story cards positioned on the wall.

4.1.1 Feedback

A small expert test of the prototype was carried out at the end of the iteration. Sandy Mamoli, an Agile Coach and the President of the Agile Professionals Network, Wellington, offered her time to test and provide feedback about the prototype. She is very familiar with Agile and story walls and as such makes an excellent candidate to provide feedback.

The main point Sandy made is that each story should have its own row or lane on the wall, as shown in Figure 2.1. A task that is part of the user story then makes its way across the lane which has the different stages. Sandy also offered input into what direction the project should take by voicing her opinion on archiving of the wall. She said that as an agile coach, she wasn't interested in the history of tasks along the wall and that is not the purpose that the wall should fill. It was more important for her that look and feel of the paper wall is preserved whilst adding an electronic element.

4.2 Iteration Two

During iteration two the wall went through two stages. The first was to refactor the wall to allow for multiple stories that create lanes on the wall. The second was to implement space utilisation features.

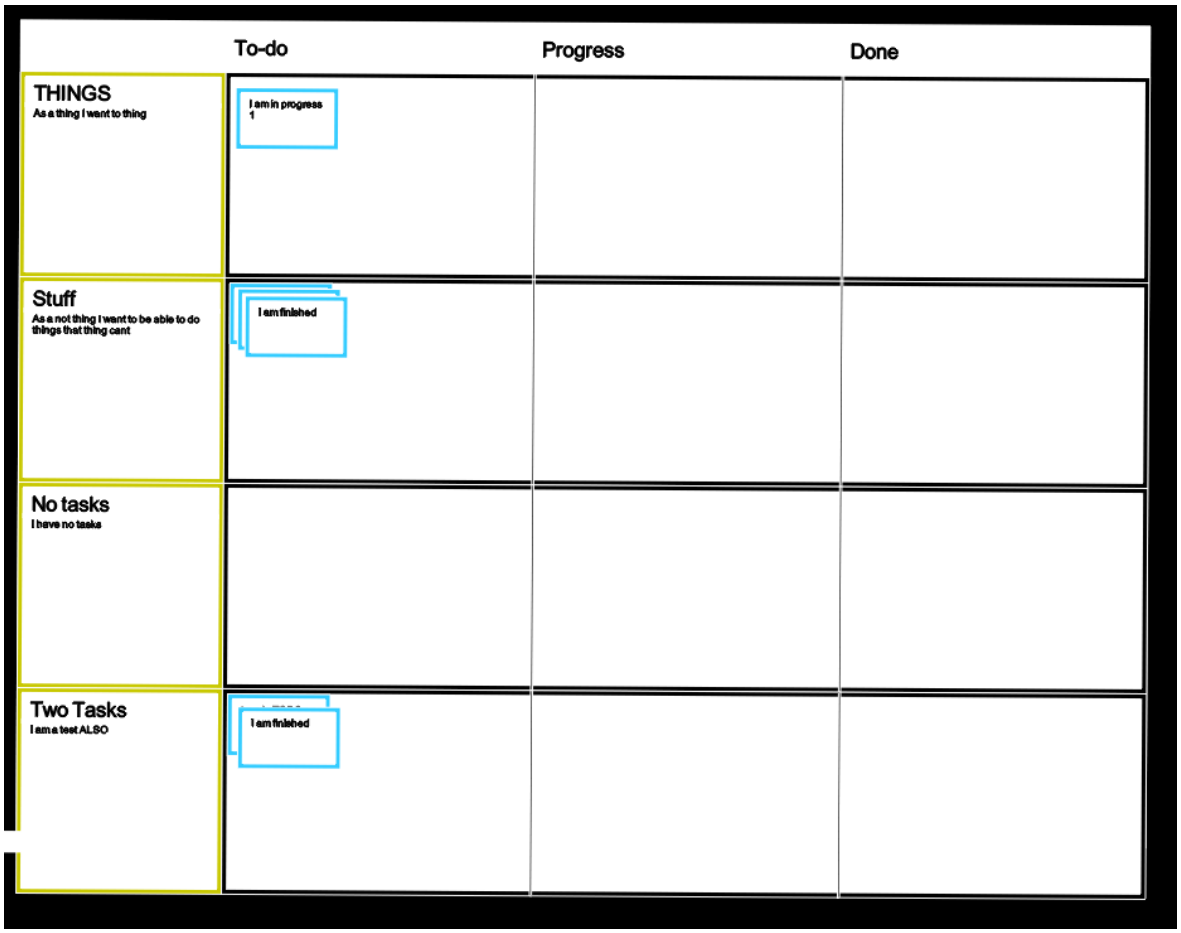

Figure 4.2: Story wall at the mid stage of iteration two.

4.2.1 Refactor

Adding of lanes

Adding support for task cards within story lanes was done over a few weeks as it required large refactoring of the initial prototype to support this feature. The wall was updated to show story cards as defining 'lanes' on the wall with task cards able to move between columns within the 'lane'. The new look of the wall is shown in Figure 4.2.

Story Prioritisation

This refactoring allowed for the reordering of story cards to be implemented. A drag gesture on the story card sees the whole lane made moveable up or down. Using vertical ordering to represent stories priority now allows the stories and their containing tasks to be reprioritised with a single flick.

To implement this drag, a new gesture was created that is a modified drag. It only allows for the object the gesture is done on to move along the vertical axes.

XML Loading of Wall

Included in the refactor was the ability to load the wall in from an xml file. Data input is outside the scope of this project and loading pre-made story walls from xml would be sufficient. The name and number of the columns are given in the wall description. This allows for customisation of the wall to support any form of wall, be it Kanban or Scrum. An example xml file is given in 4.3

```
<?xml version="1.0"?>
<wall>
        <column>Todo</column>
        <column>Progress</column>
        <column>Verify</column>
        <column>Done</column>
        <story>
                <title>Display times</title>
                <desc>As a user I can see times</desc>
                <task>
                        <desc>Add show times to database</desc>
                        <stage>1</stage>
                </task>
                <task>
                        <desc>Update view to display time </desc>
                        <stage>1</stage>
                </task>
        </story>
</wall>
```

Figure 4.3: An example XML file for a wall with four columns, one story card, and two task cards.

4.2.2 Space Utilisation Features

Now that the groundwork of the wall had been completed, it was time to explore more of the benefits that a digital story wall can provide. One benefit, the reordering of story cards, was already implemented. Previously mentioned, we focused on the utilisation of space in the layout of cards. Two features were implemented that aid in space utilisation: Providing a layout of the cards, and auto sizing of story lanes.

Layout Algorithm

The first step in utilising space is providing a layout algorithm that displays the tasks in an effective way. Each task is positioned next to the previous one until they do not fit horizontally, at this point the height of the lane is increased and proceeding tasks are placed below. This involved splitting up a lane that was previously an entity that held task cards into something that held stages. These stages then contain and manage the positioning of task cards. This is the difference between the appearance of columns in Figure 4.2 and Figure 4.4

Auto Sizing Stories

Since task positions are now enforced by a layout algorithm, the next step was to have the story cards, and their corresponding lanes, resize themselves to accommodate the task cards whilst using the least amount of space. The more task cards inside a stories lane, the larger the lane would become. This provides a quick way to identify bulky stories with many tasks, an action that would be more difficult to do on a static paper based wall. It also means that there is no physical limit to how many tasks a story can contain. Figure 4.4 shows varying sizes of stories and lanes due to the varying number of tasks within each one.

4.2.3 Feature List

The story wall at the end of iteration two had the following features, and is shown in Figure 4.4

- The wall is loaded from xml
- Each story card has its own lane containing tasks.
- Task positions use a flow layout
- Height of a story lane is determined by the amount of tasks within it
- Priority of the story cards, determined by their order, can quickly be changed

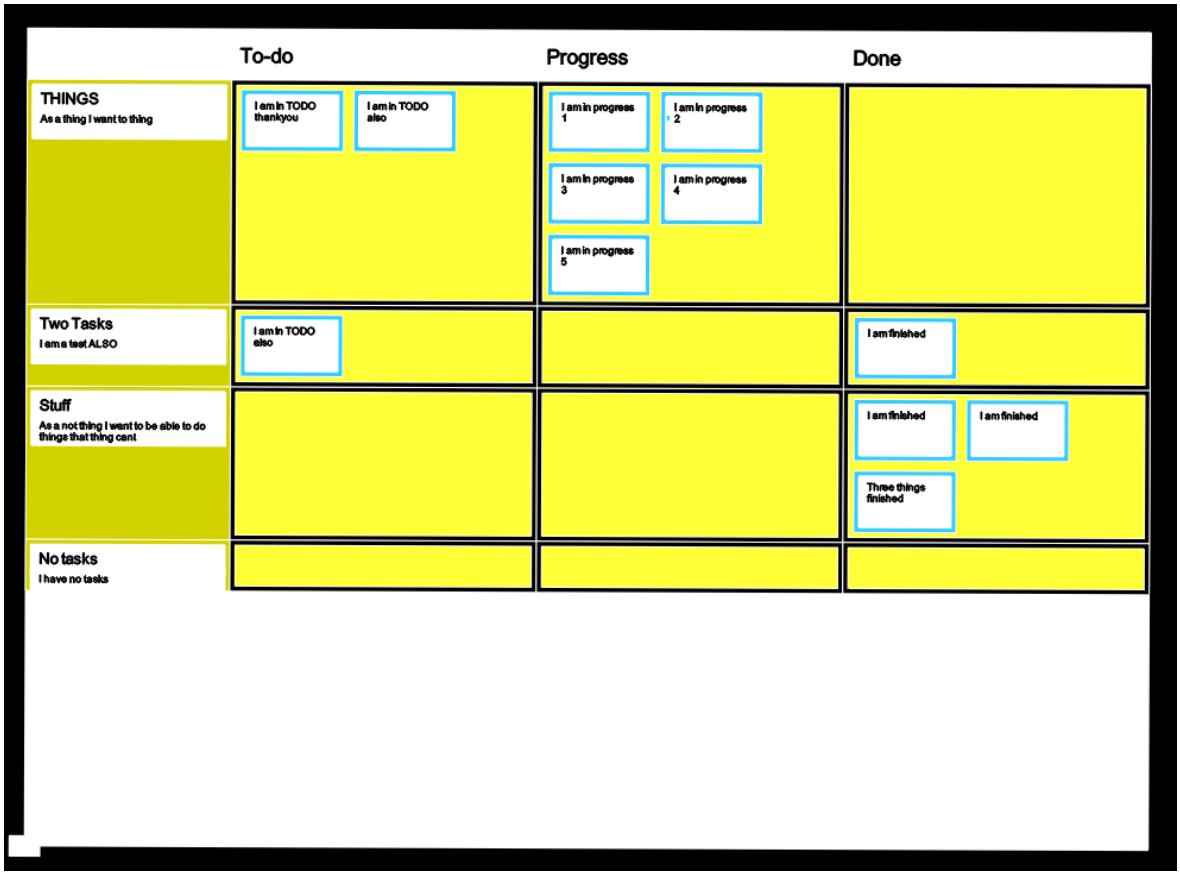

Figure 4.4: Story wall at the end of iteration two. Shows the layout of task cards (blue) sizing the story row (yellow).

4.3 Iteration Three

4.3.1 UI Upgrade

Entering the final iteration, the first thing to do was to give the user interface an upgrade. Until now, the look of the wall was not as important as getting it to function.

The first thing to do was tweak the positioning of the squares that defined the columns and rows. Each square was placed next to each other, giving the appearance of a double border. To solve this, the squares were overlapped to make it seem like there was just a single 1px border around each one. This removed the feeling that the wall was made up of pieces, and instead made it seem like each column and row were defined by a single line, not the borders of many squares.

The next thing to fix was the displaying of the story cards. Previously, the text in a story card had a white background that posed a problem because of how they resized to accommodate task cards. The story card itself could resize, but the text within it cannot resize without distorting the text, meaning the white background of the text would remain static. The result of this is an ugly looking story card that did not have a consistent look. The background of the text was set to be transparent to overcome the problem. The border of a story card was also fixed to line up with the new borders for the columns and rows to keep the consistent feel throughout the board.

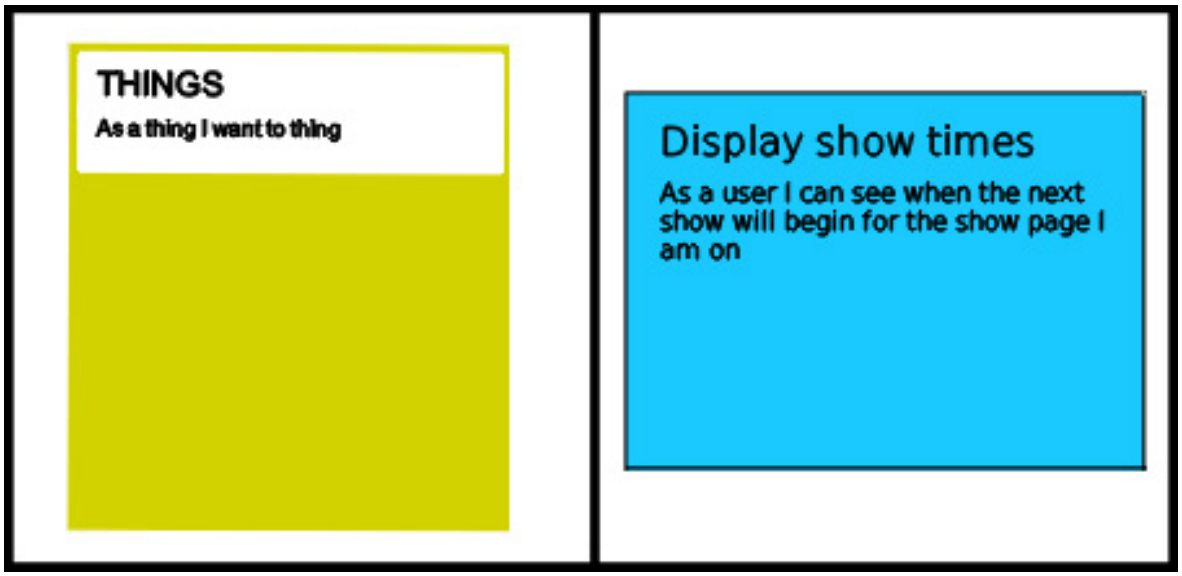

Figure 4.5: Old style story card on left. New style story card on right.

Task cards were revamped in three ways. First, the border around the card was reduced to 1px to be consistent with the story card. Secondly, task cards are given a random offset of between -3 and 3 degrees. This is to give the cards a feeling of how they would be placed on a paper wall. Lastly, the task cards have been given a shadow behind them. This is to again give them the feeling of being stuck on the wall, and able to be picked up and moved. The wall has also been given an encompassing border. This is to make the wall feel like one entity. Previously the components of the wall were floating around in free space. Adding in this border separates the wall from the empty background, giving it a feeling of completeness.

Another small change to the interface is the way task and story cards behave when you 'pick them up'. Dragging task cards around now causes them to move closer to the camera,

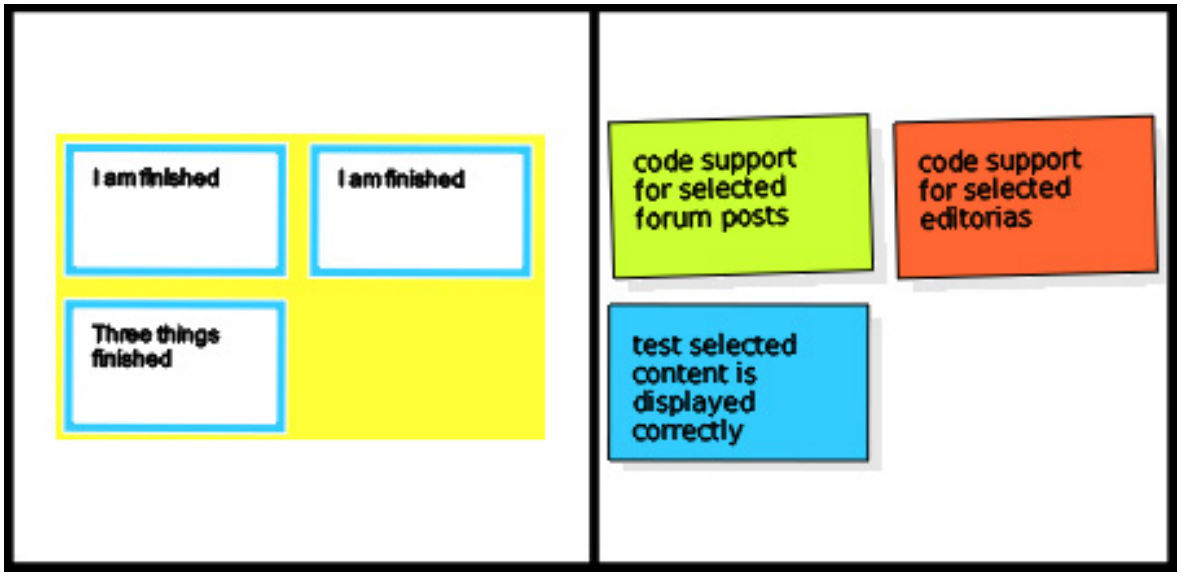

Figure 4.6: Old style task card on left. New style task card on right.

giving the user the feeling that the card has been picked up off the board. When a story card is being moved around, the whole row exhibits this picking up behaviour.

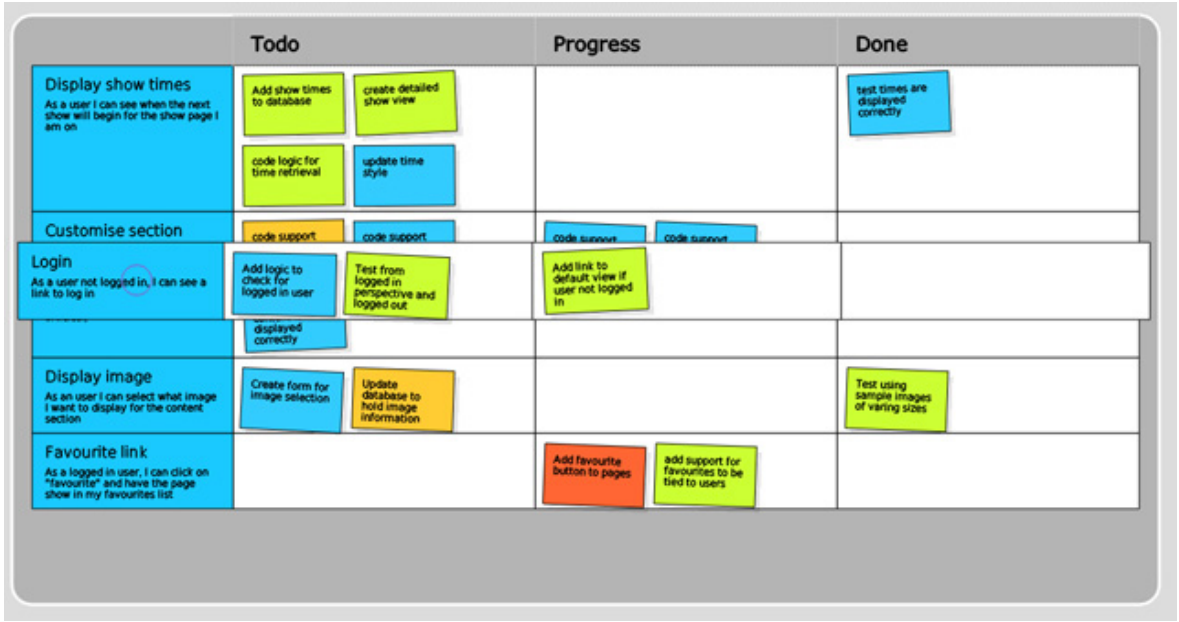

Figure 4.7: Story card being moved that is hovering closer to camera.

4.3.2 New Features

Once the interface of the wall had been upgraded, the development continued with the implementation of six new features:

- Task card resizing, incremental scale
- Colour change on task

- Colour change on story
- Semantic zoom
- Change column layout
- Rearrange tasks within stage
- Resize column width

Task Card Incremental Scale

We decided to represent the effort associated with completing a task with its size. The bigger the task card, the more effort required to complete it. Initially, this was represented with the standard pinch/zoom gesture on the task card. We decided that allowing the card to be freely scaled was a bad idea if it was to represent the effort required of the task. Without limitations on the size, it becomes hard to compare tasks and work out the difference in effort. To solve this, an incremental scale gesture was created.

Incremental scale is a custom gesture that steps up or down the size of the object 30% at a time, in this case, a task card. The standard scale gesture takes two finger positions, and when they move apart, scales the object to match the distance change between the fingers. Incremental scale snaps the object to a certain size once the distance between the two fingers has changed by a threshold. In this case, the threshold is 30%. When the fingers have moved apart by 30%, the object will snap to 30% its size. This continues for every 30% change the fingers make.

The benefit of having a task card scaled with this gesture is it provides a level of comparison between tasks. For example, if two tasks are the same size, then one is scaled up two 'snaps', it could be said that the task is two units of effort larger than the other. This kind of use would not be possible with the standard free range scale gesture.

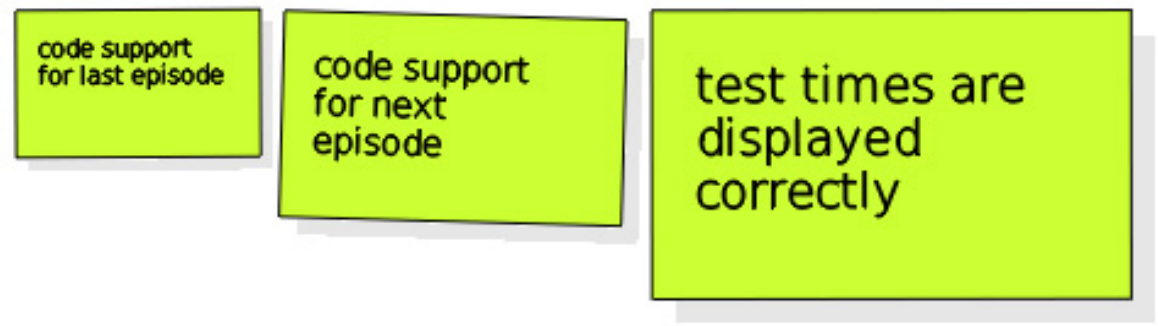

Figure 4.8: Three tasks increasing in size from left to right.

Colour Change on Task

Tapping on a task card now causes the task to change colour. It was thought that the colour of the task could represent a number of different things. Without hard coding any final decision into the prototype on what the colour of a task can represent, it leaves it open to a development team to decide meanings that work for them. This is particularly appropriate because so many development teams have their own way of doing things. In this prototype, four colours have been implemented; blue, green, yellow, and red. Taping the card causes its colour to rotate between them.

An example use for task colour is to represent the status of the card that is not shown by what column it is in. For example

- Green: Everything okay
- Yellow: Task running behind
- Red: Task blocked
- Blue: Task in testing (if no testing column present on the wall)

Colour Change on Story

The colour of story cards can also be changed with the same tap gesture as task cards. It made sense to provide this functionality if the same was also provided for task cards. The same four colours are included in the prototype; blue, green, yellow, and red.

Semantic Zoom

Figure 4.9: Furthest zoom hiding card descriptions.

The ability to zoom the camera into the board has been in place since the first iteration. A new feature added to this zoom functionality is the changing of semantic levels as the user zooms through the board. This is a feature that you commonly find on a map. The details displayed change as the zoom is changed. Less details are shown when you zoom out, giving an overview. More details are showed when you zoom in, allowing for the display of more information.

Three semantic zoom levels have been implemented, from furthest to closest;

- Blank cards
- Card descriptions
- Card descriptions + details

There are two reasons for having the furthest zoom display only blank cards. Firstly, having the board this small provides only an overview of the project. This is achieved through

information such as the amount of cards in each column, the size of each story card, and the size of task cards. The descriptions of the tasks are not important at this level. The second reason for blank cards is an issue with the way MT4J displays text. Using vector fonts to support scalability, the text becomes unreadable when it gets too small. This problem could be avoided by using bitmap fonts, but then the board would not respond well to zooming at all, resulting in distorted and blurred text. The solution was to hide the text when it became unreadable. This boundary of readable/unreadable text became the threshold for the topmost semantic level.

The second level, card descriptions, is the normal operating semantic level. This shows the task descriptions, but no other details. This avoids over cluttering of the task cards with information that is not usually needed when interacting with the wall.

The third, and closest level, displays the task descriptions along with extra details known about the card. For this prototype, the extra detail added to the task is the date it was added to the board.

At the closest level, the font size of the description are also reduced, this is to accommodate large descriptions. If a description is too large for a card, the overflow of it is not displayed. The font size could not be reduced at the standard semantic level because of the vector font sizing issue discussed before. Zooming into the board displays larger cards which also have larger text. Reducing the font size to match the smallest displayable vector font size now creates a much larger area for text to fit in. This allows long descriptions to fit on the task card whilst maintaining a readable font size.

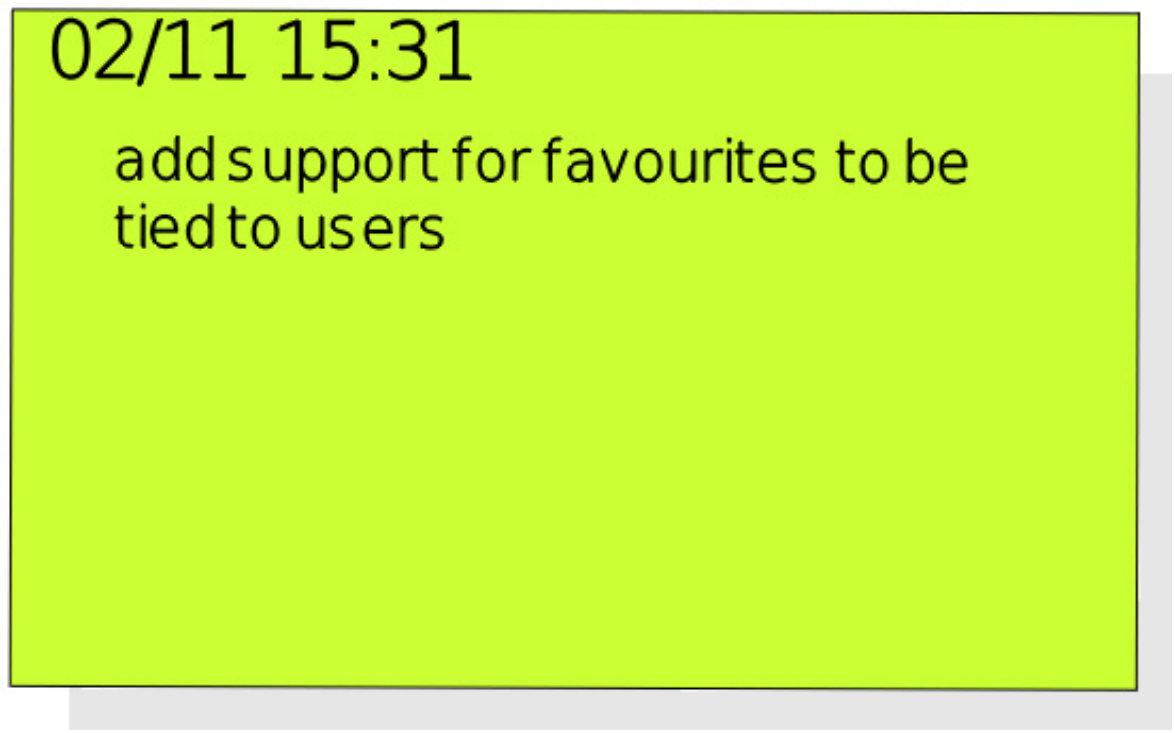

Figure 4.10: Closest zoom showing story card details.

Change Column Layout

Touching and holding the top of a column for two seconds changes the layout algorithm used for tasks within the column. The new layout created 'stacks' the cards on top of one another and reduces their size. The motivating example for this feature is tasks within the done column. Once they have been completed, they are less important than the other tasks. Sitting in the done column, they are pushing up the size of the stories they belong to. Using this stack feature makes the done column have less of a footprint, the size of stories are now determined by tasks in other columns, tasks that are not done. This provides a way to use space more effectively and hide information that is not as relevant as others.

To make sure the amount of cards contained is still visible, each one is offset by five pixels. This makes it so they are all still visible, but only slightly, giving you an impression of how many are stacked.

Tasks can be easily un-stacked and returned to their previous layout with the same touch and hold gesture. All positioning and sizing of the cards is preserved when returning to the standard layout. It was important to preserve layout and sizing because these attributes have may meaning on the wall to the development team. This meaning should not change because the tasks were stacked and un-stacked.

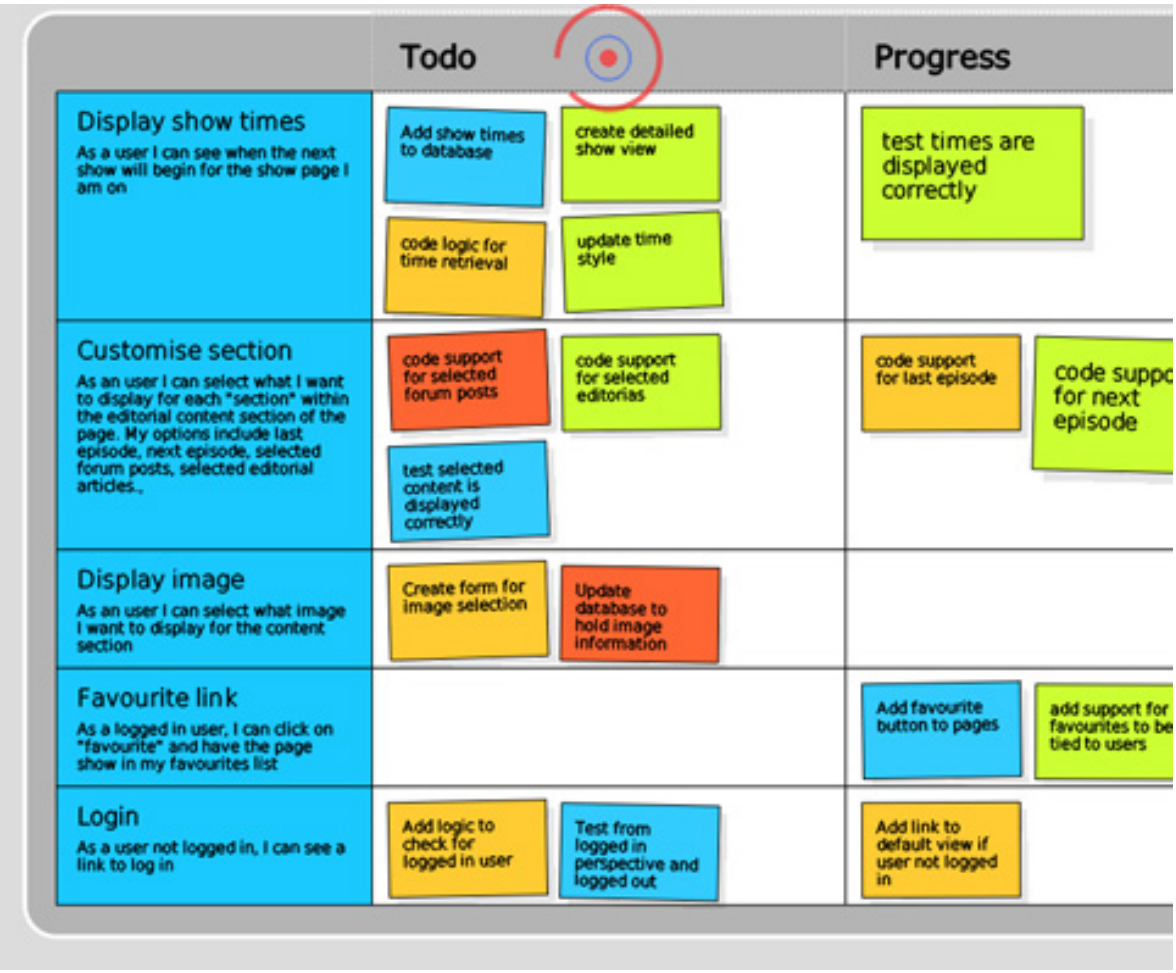

Figure 4.11: Stack gesture mid way though. Shown by red progress circle.

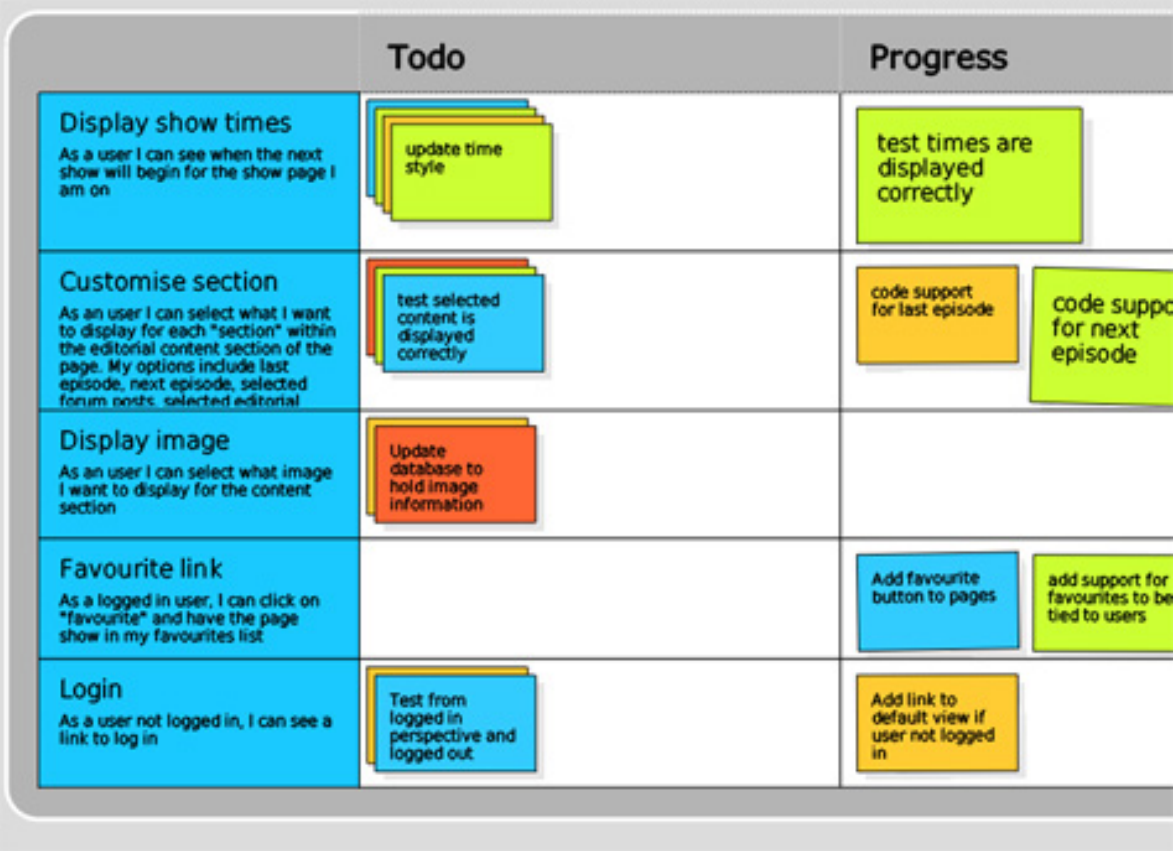

Figure 4.12: Stories stacked up once stackgesture completed.

Rearrange Tasks Within Column

The standard layout algorithm was upgraded to allow the reordering of tasks within their current column. This is done by dragging a task and releasing it on top of another. The task underneath will shuffle over to the right, causing other tasks to shuffle over as well. The dropped task will take over the position of the task it was dropped on.

This feature gives the ability to add meaning to the ordering of tasks. For example, the priority of tasks could be determined by the ordering within the column. This priority can be easily changed by dragging the card around.

Resize Column Width

We wanted to be able to resize columns quickly with a single gesture. Initially we discussed a gesture similar to scale that could be performed on a column to accomplish this. The problem is that a pinch gesture on a column would be interpreted as a pinch gesture on the board, something that is used for zoom. There was no way to have the same gesture to both zoom and resize a column. To solve this, the gesture is preformed on the top section of the column that holds its name.

A custom gesture was created to achieve the column resizing. It is a modified scale gesture that ignores the distance in the vertical positioning of the fingers, and only takes the horizontal. Instead of sending an event to the object the gesture was performed on it sends it to the wall to resize the column. This sees the column stretch out to match the distance that the fingers separate by horizontally.

4.4 Summary

The final prototype is shown in Figure 4.13. It has the following features:

- The wall is loaded from xml
- Tasks can be dragged along and dropped into columns
- Height of a story lane is determined by the amount of tasks within it
- Priority of the story cards determined by their order can quickly be changed
- Task card can be resized to reflect their effort
- Colour of tasks are changed with a tap
- Colour of stories are changed with a tap
- Semantic zoom to show and hide extra details
- Task layout algorithm can be changed for an entire column, utilising space
- Tasks can be re ordered within columns
- Columns can be resized

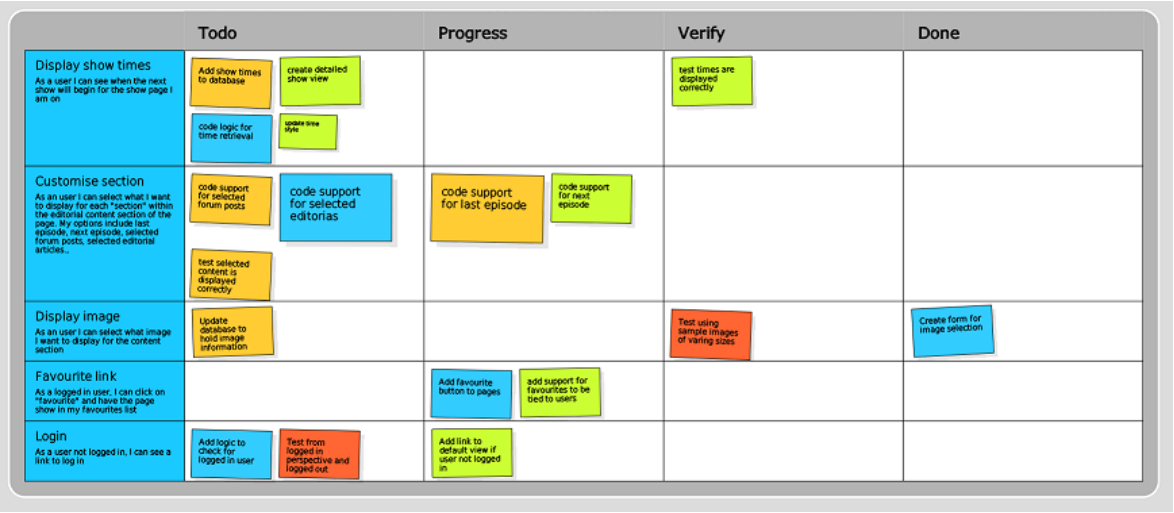

Figure 4.13: Final prototype.

Chapter 5

Implementation

This chapter describes the technical details of MT4j and provides an overview of how it was used to create the story wall prototype.

5.1 MT4j

MT4j is an open source framework for the development of multi-touch applications. The version used in this project was 0.98Full.

MT4j provides two main features for creating a multi-touch application, gesture recognition, and a component hierarchy.

5.1.1 Components

Components in MT4j make up the interface of an application. A hierarchy of components is created by attaching components to each other in a parent/child relationship. This allows complex structures to be built up using components. When a component is transformed, all the child components in the hierarchy also get the same transformation. This allows a composite of components to be treated as a single entity. Some example transformations include; scale, move, and rotate. Component positions are represented in a three dimensional space with x,y, and z values. There are three different positions a component has; position relative to self, position relative to parent, and global position. Changing the position relative to self will translate the component without performing the translation on its child components. Changing the position relative to parent will offset it from the parents position. The global position is its absolute position in the scene.

5.1.2 Gestures

MT4j offers a range of gestures that can be attached to components. The three most basic are:

Drag Moves a component around by touching and dragging it

Rotate Rotates a component by a two fingered rotation motion

Scale Scales a component using two fingers to 'pinch' it

These gestures are registered to components of your choice. MT4j takes care of recognising and capturing gestures, then sending gesture events to the component that perform translations on themselves to reflect the action.

5.2 Final Prototype

The story wall is made up of four components; MTWall, MTStoryCard, MTStage, MTTaskCard. These all extend from the MT4j component MTRectangle, giving them their drawing methods. Sections 5.2.1 through 5.2.6 refer to elements on the class diagram given in Figure 5.1 on page 30.

5.2.1 MTWall

MTWall contains three things; A storage object, ColumnHeads, and MTStoryCards. The storage object is responsible for creating Story and Task objects from xml, and giving them to the wall to create MTStoryCards. The wall is responsible for laying out the MTStoryCards along the y axis every time they change height or position. The wall has two gesture listeners; drag and scale.

5.2.2 MTStoryCard

MTStoryCard has a modified drag gesture registered on it where it removes the x value from the gesture event before it is fired. This results in a card that can only be moved vertically. MTStoryCard objects contain MTStage objects. These are children of the story so that when the story is moved, the stages move along with it. Stages are positioned next to one another to form a row. The number of stages depends on how many have been specified in the xml file. Stages have had gestures disabled on them. This stops the stage from catching gestures and passes them up the chain to a parent, in this case the wall. This allows the walls scale and drag gestures to be done on top of the stage but still be caught by the wall.

5.2.3 MTStage

MTStage objects contain MTTaskCard objects. MTStage is responsible for laying tasks out and informing its parent, the story card, what height it should be. This is achieved by each stage telling the story card what the minimum size it must be in order to contain tasks. The story then sets the size of all stages to the largest minimum size. The layout method is called every time a drag gesture on a task card ends over the stage.

5.2.4 MTTaskCard

MTTaskCards have two children components; a text area and a shadow. The task card has the custom created incremental scale gesture registered on it. The task card is responsible for telling its parent story card that it has moved. The story card then finds the stage it is positioned over, and adds it as a child of that. If no stage is found, the story asks the wall to see if it was dropped over a different story card. If it has, it adds it as a child of the new story.

5.2.5 Columnhead

ColumnHead is an invisible MTRectangle that has the custom column scale gesture on it. It has one component as a child, a text box that has to display the column title. This column head serves the purpose of allowing columns to be resized. The custom scale gesture only takes into account the change in horizontal distance between two touch points, and sends an event to the wall. The wall then tells the stories, which contain the stages, to resize the corresponding stage.

5.2.6 LayoutStrategy

The layout algorithms for task cards are encapsulated used the strategy pattern, allowing the layout to be changed at run time. This allows the user to change the layout of cards between the stacking layout and the flow layout.

5.3 Difficulties

A major difficulty working with the MT4j framework is the inability to pass gestures up the hierarchy of children and parents. A component can have gestures disabled on them by setting setPickable() to false. When a component is un-pickable, its parent catches gestures instead of itself. In the case that its parent has a drag gesture registered, it would be possible to drag on the component and have the parent move around. If the component was a button however, it would need gestures to be enabled. The component would respond to tap gestures, but it would not respond to the drag gesture of its parent. The drag gesture would not be registered to the button, so a drag would not be caught. This results in a dead spot on the parent for the drag gesture.

This is the reason for the gesture placement to resize columns to be at the top on the ColumnHead object. If a scale gesture was present on the MTStage object, which makes up columns, it would not be possible to have a drag gesture on the entire board. These stages would be enabled to gestures but only have the scale registered. When a drag gesture was done on them to move the wall around, they would not respond because only a scale gesture is allowed. The proper behaviour should be that if a gesture is not registered to a component, it should consult its parents to check if it is registered there. If it is, its parent should handle the gesture. This would prevent gesture dead spots from appearing. This kind of gesture handling is not present in version 0.98 of MT4j, but is set to be implemented in future a release.

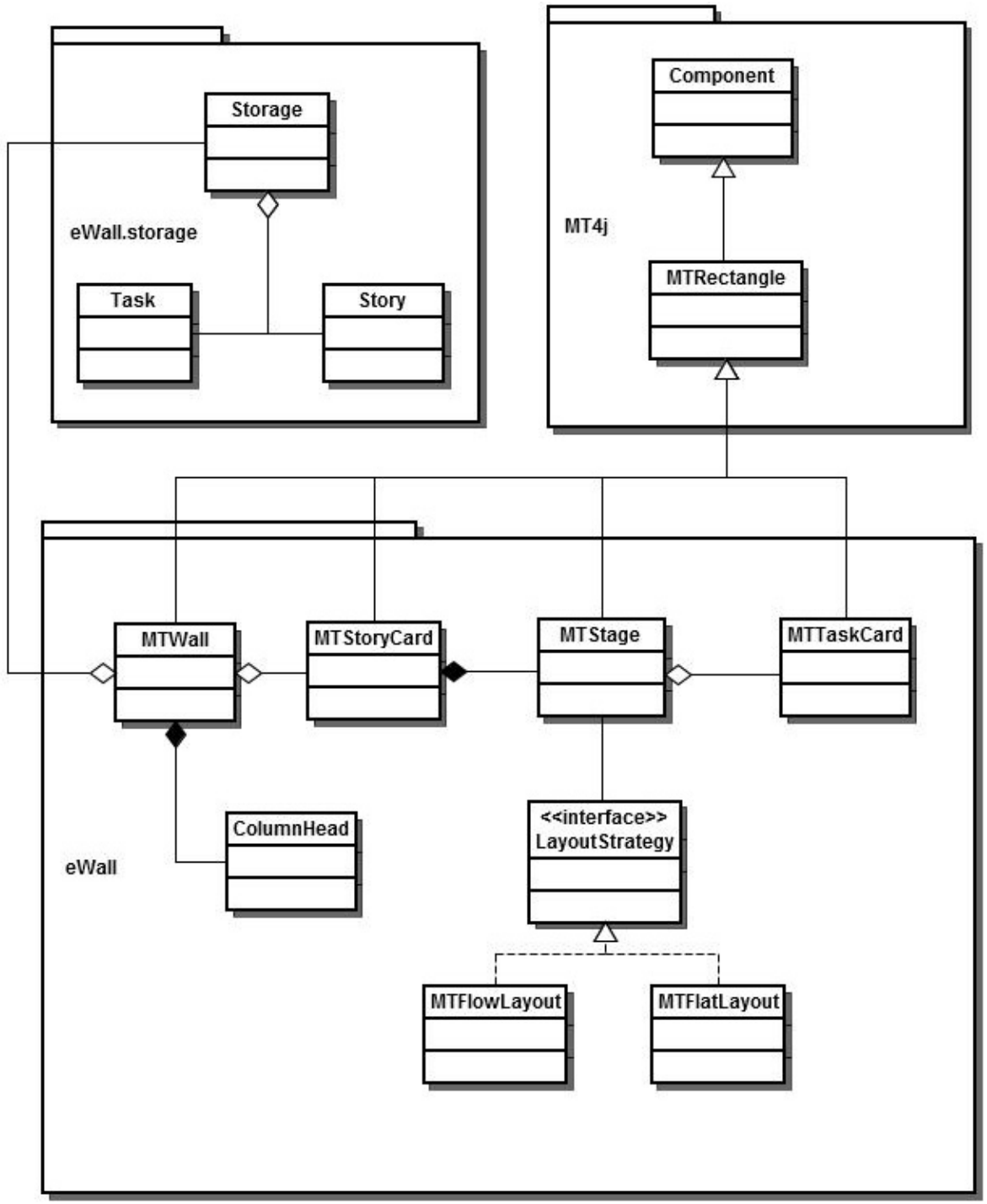

Figure 5.1: Class diagram for prototype.

Chapter 6

Evaluation

The nature of the story wall allows for a qualitative experiment to determine the effectiveness of the features it provides and investigate how users would interact with the wall. A full study on a development team using the wall would provide useful results but is outside the scope of this project. User testing best fits this project as each one would take only an hour, and see experts use the tool and provide their feedback. This gives an insight into what might happen in a real development team without having to conduct a long term study into the use.

Three evaluations were carried out to test the wall:

- User evaluation
- Demonstration for International Researchers
- Heuristic evaluation

6.1 Test Scenario

For both the user test and heuristic evaluation, a sample project was created and the wall was set up to reflect it. The project was website booking system. The current iteration featured five story cards. Each story card had a varying numbers task cards, making sure to have a mixture of colours and sizes. The scenario was consistent through the user testing and heuristic evaluation. This is to give the same hints of possible gestures to perform through the varying sizes, colours, and placement of the cards. The setup of the wall is given in Figure 6.1

It is possible to compare and contrast the results of the user evaluation with the results of the heuristic evaluation because they use the same test scenario. It would not be a fair comparison if the heuristics were preformed on a different set up, and it would not be fair if each user had a different setup.

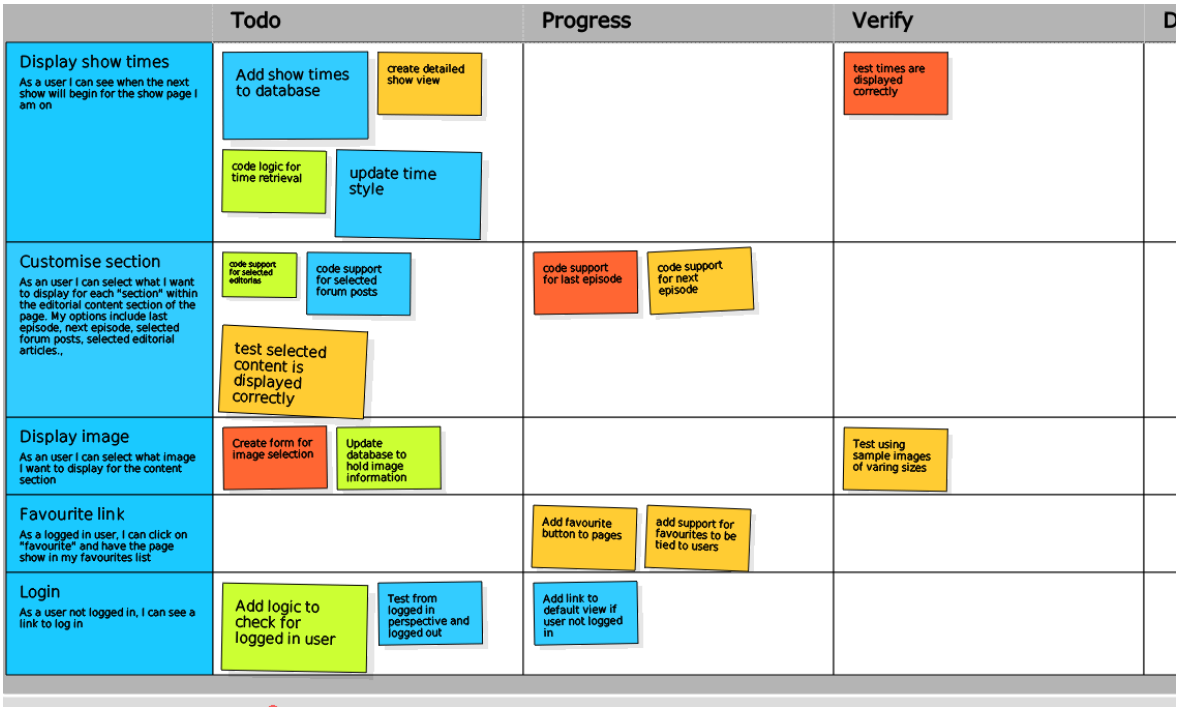

Figure 6.1: Story wall with sample project.

6.2 User Evaluation

6.2.1 Participants

Three experts and current Agile practitioners were brought in to undertake the evaluation.

Participant	Experience with Agile and Story Walls	Touch device experience	Previous walls used
P1	20 years	Yes	Paper, Leankit Khanban
P2	7 years	Yes	Paper, AgileZen, Custom implemented wall
P3	9 months	Yes	Paper

Table 6.1: Table of domain experts.

There is more than 27 years experience with Agile and story walls between the participants. This provides an excellent means of evaluating the wall with domain experts who are all target users of the prototype. All three participants have used a touch device before, and P1 and P2 have both used some form of electronic wall before. None of the participants have used a touched based wall.

6.2.2 Procedure

Each test lasted approximately an hour and followed a list of use cases for the participants to carry out. A use case comprised of a small task to be performed on the board, such as moving a card between columns. Use cases were asked to be preformed, were indirectly done through the actions of another, or completed through the participants exploration of the board.

The following nine use cases were evaluated:

UC1 Moving tasks between columns

UC2 Changing task layout, 'stacking up cards'

UC3 Changing the effort of a task by changing its size

UC4 Re ordering story cards

UC5 Re ordering task cards

UC6 Resizing a column

UC7 Using semantic zoom to find details about a card

UC8 Changing a tasks colour

UC9 Changing a stories colour

For each of UC1 - UC9, four questions were addressed when applicable:

Q1 Intuitiveness of the feature: Can they complete the action without being told how to do it?

Q2 Usefulness of the feature: Would they use it?

Q3 Alternative use of the feature: What would they use this feature for?

Q4 Comparison with paper-based system: How would they do it on a paper counterpart?

Six general questions targeting the following were asked at the end to spark more specific discussion around certain ideas:

GQ1 Standout features

GQ2 Useless features

GQ3 Missing features

GQ4 Intuitiveness of interaction

GQ5 Preference of prototype over a paper wall

GQ6 Maintaining the 'feel' of a paper wall

Participants were encouraged to play with the wall and provide any feedback at any time. Discussions about the board were uninstructed and started from the list of questions above. The user tests were recorded with the participants' consents to provide the best recall of feedback given.

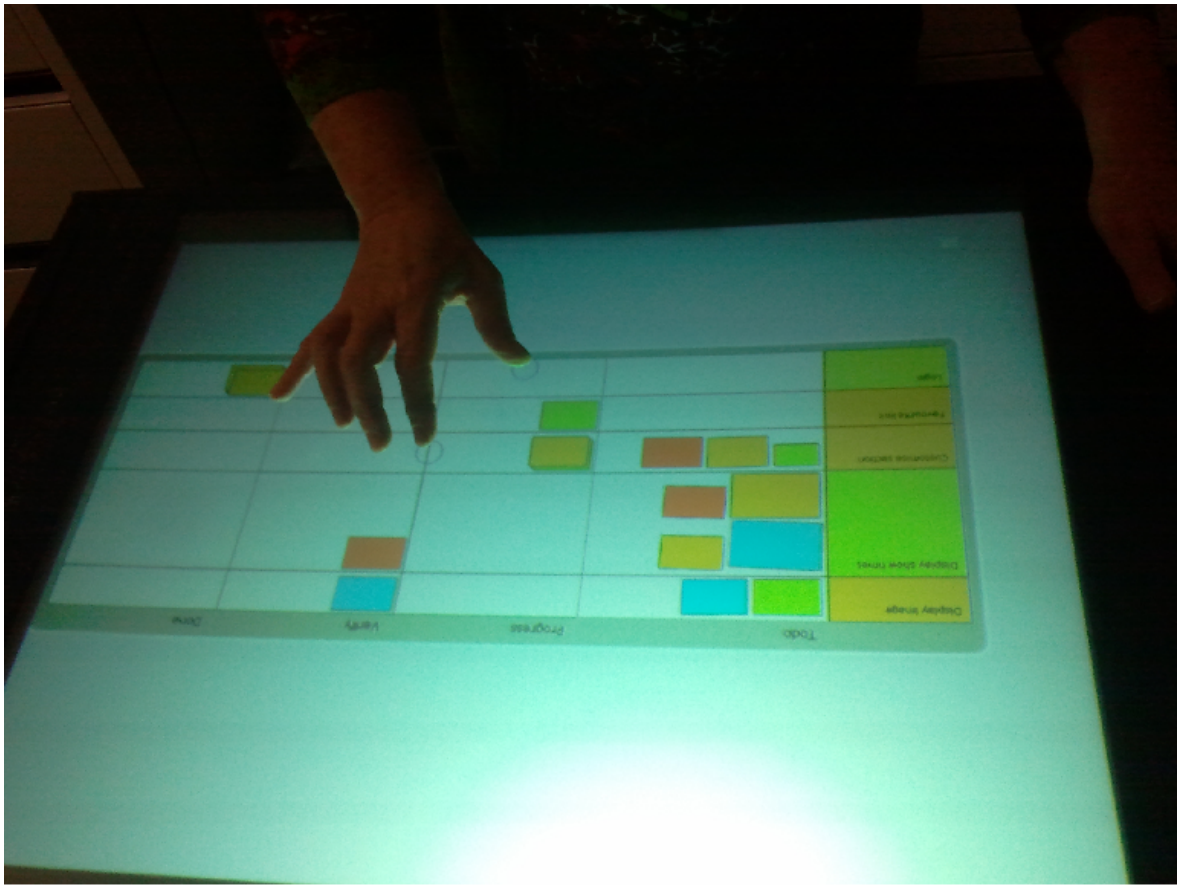

Figure 6.2: Participant performing evaluation.

6.2.3 Results

UC1: Moving tasks between columns

When asked to move all the tasks of one story into 'done', all participants had no problem in instinctively touching a card and dragging it into place. This suggests that the action of moving cards around on a touch table is intuitive to our participants as they have all had previous experience using a touch device.

P1 noticed that they could drag a card across into another stories lane without being challenged to make this move. This is an area where a mistake could be made and without the wall bringing it to attention, it could be done by accident. There would be no way of knowing where the card came from.

P2 felt that when they released the card, they did not like how the story wall automatically positioned the card within the section. They felt it would be better to have ultimate control over where the card was place, but perhaps have it as a feature that could be turned on or off.

UC2: Stacking up the cards by changing their layout

Participants were all asked to 'stack the cards on top of each other' for the done column. As expected, they all picked the cards up and tried to drop them on one another. This resulted in the cards been rearranged within the section. No participant could figure out how to change the layout of the column to stack the cards up without been told. P1 and

P2 both said it would be good to have an icon indicating that the top of the column could be interacted with in order to affect the layout of the tasks within it. Once familiar with the gesture, they appreciated that it was easier than the more intuitive 'pick and drop cards together' action they tried before.

When queried on the usefulness of this stacking functionality, all three participants said that it was a useful feature. P1 and P2 both went further by mentioning that it's a good function to handle complexity of a large project, as it reduces the space that tasks take up on the board, but there needs to be a way to only un-stack a few selected stories at a time. In large projects un-stacking a column would increase the size of the board, making it hard to find the story that has the tasks the user was interested in.

UC3: Changing effort of a task by manipulating its size

Participants were asked to take a task card, and make it reflect that the task had now doubled in effort. P1 and P2 went straight for the scale gesture, whilst P3 first tried tapping on the card, and then used the correct scale gesture. All three participants increased the size of the card one step.

P1 and P2 both noted that size was missing a sense of scale. They said that there needs to be a frame of reference for what the size of the card is, and what that means in real terms. P1 said there should be some implications for making the size and effort larger. For example, only allow 8 hours of work each day and making a task bigger would take up more of the allowed hours of work.

P2 said that size reflecting effort in this way would work for some teams, but not for others. The reasoning is that in a previous team, they used effort scale of 1-4. This would work having the tasks between 1-4 steps difference in size. In the current team, they use an effort scale of 1-30. Using the current scale gesture, tasks would become too large if they were 30 steps of size difference.

P1 and P3 both said that this is what they would use to represent effort with. P2 thought it would better represent importance. Effort of the task is something that is only important during planning stage, and is used by management to determine how much work they think they will select to be done. Once a task is on the wall, it's used by the team, not management. Having a card be bigger in size draws attention to it. To P2, this attention should be given to a task that has more importance, signalling to the team to get it finished, rather than signalling that the task will require more effort to finish it.

All three participants said they represent effort with a number written on the task card. In the unlikely event that the estimated effort was changed, it would be crossed out and rewritten.

UC4: ordering story cards

Participants were told that the priority of a story is represented by its vertical ordering and to make one story card a higher priority than another. P1 instinctively grabbed the story card and shuffled it. P2 first grabbed the row and tried to move it, then grabbed the story card. P3 first tapped on the story card, then grabbed it and dragged it to the correct position.

On a paper wall, P1 and P3 said that story cards are given priority based on their vertical ordering, and when the ordering changes, the cards are moved by hand, which can be a pain. P3 said that there is no relative priority given to a story inside an sprint whilst using scrum, or if using Kanban, there are not many cards on the wall and priority is irrelevant.

P1 and P3 said that this is a useful feature. P2 said that it was an interesting feature as teams may start to prioritise cards within a sprint as it is so easy to do, it may form habits.

UC5: ordering task cards

Participants were asked to rearrange the tasks within a section to reflect a change in priority, represented by their position. All three participants had no trouble shuffling the cards around.

P2 said that the priority of tasks is not that important, and size would be a better indicator of priority if it was needed. When using a paper wall, they said that you do not think about where you place it, but because the layout of cards is enforced, teams may start to utilise the positions of the tasks to signify priority.

P3 said that the tasks are not formally prioritised on the wall, but the team informally knows what tasks are more important. It is often not possible to prioritise cards this way on a paper wall because there is not enough room on the wall. They said that having this feature could be useful, especially if you are removed from the team and do not know a tasks informal priority.

P1 used vertical ordering of the cards to represent priority, but was not very strict on it. They also said that colour is another way to show priority.

UC6: Resizing column

Participants were asked to make a column larger. All three participants instinctively tried the scale gesture on the column, causing the board to zoom in. None found the correct gesture without been told. Once they knew that they could resize the column using the top of it, all three still had trouble using the gesture. The common action was to touch the edges of the column, and attempt to drag them out. Once the gesture was clarified, participants could execute it. P1 said there should be some indicator at the top of the column which shows a gesture is possible.

All participants said this is a useful feature. On a paper wall they said space runs out frequently, cards have to be stacked up on top of each other, leaving only a few showing. P3 said they purposely constrain themselves not to have too many cards, but removing this constraint would be good. When queried if the constraint helps them limit the amount of work in progress, they said that it is a not an issue, and just a physical constraint.

UC7: Using semantic zoom to find details about a card

Participants were asked to find out more details about a task card. This was to test how intuitive a semantic zoom would be to provide more details. All participants had previously discovered the zoom gesture on the board before being asked to perform it. No participant used the zoom gesture to find out more details about a card. The instinctive action they all did was tap on the card. Once they saw that this gesture changed the colour, they tried tap and hold. P2 tried the correct zoom gesture, but did it on the card it, not the wall, causing it to scale the card size.

P1 and P2 both said that using semantic zoom to display details is a bad idea. P1 did not like having to zoom in to the point where only one card is visible in order to find more details about it. They suggested that the tap gesture would be a better fit. P2 said zoom would not be too bad if the table surface was more responsive and similar to a smart phone but most details about a task card needs to be visible at all times. They suggested having time on the wall, ownership, and status all visible and notes as extra when the task is tapped. P3 liked the idea of the semantic zoom but thought that they had to zoom in too far for details to appear. They would have liked the boundary between the zoom levels to be smaller.

UC8: Changing a tasks colour

Participants were asked to change the colour of a task card. No one had a problem with this because they had all previously discovered the tap gesture to change colour, often not on purpose. Throughout the testing, the colour was continually changed by accidentally tapping on a task whilst attempting to perform another gesture. All three participants found this frustrating, especially as there is no undo, and it has to be cycled back to the original.

P1 and P2 both used colour to represent some kind of status or type of card. They said that it varies team to team. As such, both suggested the selection of colours to be customisable and have a legend displayed to cement the meanings of the colours team to team. P2 said that the colour of the task is changed all the time, in which case it is ripped off the wall, and a new one written up. It is good that the colour can be changed easily, but not too easily. It needs to be made mistake proof and have undo features. Participants were particularly concerned about changing a task colour without noticing it, and having large implications for the development team.

P3 said they colour tasks on a paper wall to correspond to what story they belong to. It was useful because if a task falls off the wall they know what story it came from. When asked if they would still want this on a digital wall where tasks could not fall off, they said they would probably still colour the tasks to match the stories for visual appeal.

UC9: Changing a stories colour

Changing story colour also suffered the same criticism of accidental touch gestures causing the colour to change with no mistake proofing or undo feature. P1 said that colour is used to represent the domain of the story card. P2 said they previously had not coloured their story cards, but it would be more useful in the product backlog than the wall to represent domain. P3 said they would use the colour of story card to represent the entire story as being blocked.

GQ1: Standout features

P1 said the standout features they liked about the wall were that it was interactive, and tasks cards can be dragged along the wall. They also said that changing the story cards order with a drag gesture would be really useful.

P2 said that they liked the way tasks can be easily staked up to use space efficiently. They also liked the idea that with further development, the table could provide a solution with all management tools in once place.

P3 said they liked the whole wall being on a multi-touch table, without singling out any feature they thought stood out over the rest.

GQ2: Useless features

When asked if there were any features they would not use, P1 said that the loved that the colour can be changed, but they would not use it because the gesture to do it is a pain.

P2 said they would not prioritise story cards, making the ability to change their order useless. They also said that although they would not use it, other teams may.

P3 said that there were not any features that they would not use. If features are made available, people tend to find a use for them. They said that the use may be different than intended, but it is good to have versatility.

GQ3: Missing features

Common missing features all three participants asked for were a product backlog and archive or metrics. They all suggested that a swipe to the left should open up the backlog, and a swipe to the right should open up the archive and metrics. Avatars to indicate ownership of cards were also requested by everyone.

P1 would have liked the board to be more dynamically customisable. This includes being able to edit, add, remove, and rearrange columns at run time. They also said a search feature would be good, along with tags on cards so they can be filtered to only display relevant ones.

P2 would like more details to be shown about the card from a touch gesture instead of a semantic zoom. They said these details should be customisable as they would be very team specific.

GQ4: Intuitiveness of interaction

All participants felt that the actions were generally intuitive. Although they did not instinctively find some of the gestures, once they were told how to do it, they said they would not have trouble replicating it. P2 said that they liked how the system would need no more than one page of instructions.

GQ5: Preference of prototype over a paper wall

P2 said they would only prefer this over a paper wall if they had to use it with a distributed team and it was a supported feature. Their choice was motivated by a strong inclination and habit to use paper-based walls over electronic ones.

P1 and P3 both said they would prefer this over a paper wall. P3 said that it would appeal to teams because it removes the feeling of being back at school with having to write things on the board, and technology enthusiasts would love to use it.

GQ6: Maintaining the 'feel' of a paper wall

All participants felt that the prototype maintained the feel of using a paper wall.

6.3 Demonstration for International Researchers

Two international Agile researches at The Open University and authors of[9], Prof. Helen Sharp and Prof. Hugh Robinson, and a PhD student researching Agile at The Open University, Jennifer Ferreira, were given a demonstration of the prototype.

The demonstration was conducted over Skype. The time difference resulted in it being given at 8pm NZ time and 8am UK time. There were technical difficulties getting it set up so that the table was visible through Skype. In the end, a camera was held above the table whilst the demonstration was given.

The demonstration lasted approximately 15 minutes whilst the main features of the prototype were shown and talked about. During this, the researches offered their feedback on some of the features and their questions were answered.

6.3.1 Feedback

All participants viewing the demonstration liked the feature of stacking up tasks within a column to un-clutter it. The utilisation of space was discussed and they felt that it would be useful for development teams. The ability to change a cards colour was discussed, and said

to be useful to represent the status of the card, but during the demonatration the accidental colour changes were noticed and stressed that it was too easy to make mistakes that could have effects on the development. They were pleased that prototype captured the overall use of a story wall, and agreed that with further development it could make a useful tool for development teams in industry.

A discussion in the end of the demonstration speculated that this would provide a real benefit to distributed teams if it had the capability to support them. They believed that this is a step in the right direction to bridge the gap between paper and electronic walls.

6.4 Heuristic Evaluation

A heuristic evaluation was carried out to assess the user interface. The heuristic evaluation identifies usability problems present in the design of the wall. This complements the user evaluation that used domain experts with participants familiar with interface design. This evaluation does not concern any area of usefulness or preference like the user evaluation, but focuses on usability only.

6.4.1 Procedure

Two 4th year HCI students and myself ran through Nielsen's 10 user interface heuristics and ranked each one low, medium, or high. We then deliberated together and decided on an overall rank for each heuristic. The heuristics were applied to the board as a whole instead of for individual use cases because of the nature of interaction with the wall. Each use case is a simple one step action and it made sense to evaluate the wall as a whole instead of small actions where most of the heuristics would not be applicable.

Nielsen's ten heuristics[7] are given on the next page in table 6.2.

Visibility of system status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
Match between system and the real world	The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
User control and freedom	Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
Consistency and standards	Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
Error prevention	Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
Recognition rather than recall	Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
Flexibility and efficiency of use	Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
Aesthetic and minimalist design	Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
Help users recognize, diagnose, and recover from errors	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
Help and documentation	Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Table 6.2: 10 user interface heuristics

6.4.2 Results

- Visibility of system status: HIGH
  The status of the system is represented by the position and colouring of the cards on the wall. These are always visible so the user will maintain a good level of feedback about the system status. When a gesture is being carried out, the user can see feedback instantly as the object responds to how the user is manipulating it.

- Match between system and the real world: HIGH
  The board is designed to look like a paper wall. This creates a high level of matching to the real world for people that are familiar with the domain of story walls. Some interactions mimic those of the real world such as dragging a card between columns, but some actions such as tapping on a card produce results that would not reflect the real world and are surprising. This heuristic is still rated HIGH as the board looks like a paper one, people familiar with the domain will have no problem using the wall.

- User control and freedom: MEDIUM
  Being on a multi-touch table, users have intuitive control and freedom over the board. There are no menus or multi-stage actions that users could accidentally get stuck in. The problem is there is no undo or redo functionality supported for changing the state of the board. If a user moves a task into a new column, the only way to undo the action is to remember where it came from and move it back by hand. This poses a problem and reduces the overall heuristic to medium instead of high.

- Consistency and standards: HIGH
  The wall stays consistent in its interaction. For example, tapping on a task card produces the same action as tapping on a story card. The gestures implemented on the wall are mostly universal gestures that are commonly used on other touch devices. Pinch zoom, drag, and scale are all gestures that you would commonly find on other touch devices and expect them to work on the multi-touch story wall.

- Error prevention: LOW
  There is currently no error prevention on the wall. An example of an error would be moving a task card from one story to another without intending to. There is no attempt from the system to warn the user that they are about to cross the boundary of a story card or limit them from making this potential mistake. Another example is the colour change gesture. During user evaluation it became clear that the colour of cards were continually changed accidentally, the system needs to prevent this error from happening.

- Recognition rather than recall: MEDIUM
  Usage of the wall does not involve more than one step actions. This requires nothing of the user to remember between dialogues. Of the actions available to the user, some are intuitive and require no instruction, for example, dragging, scaling, and zooming. Other actions such as changing task layout do require some form of prompting from the system to show the user that it is possible. The system has no instructions available for use other then the implied interaction a multi-touch table gives.

- Flexibility and efficiency of use: HIGH
  There is a high flexibility of use given the fact that novice users may only use the wall to move task cards between columns without using the extra features available. Expert users could make use of additional features like stacking up cards, changing

story order, and changing colour of cards. Although this does not speed up actions, it does provide extra functionality to expert users.

- Aesthetic and minimalist design: HIGH
  The board is designed in a minimalistic way with no menus. Only essential information is displayed. The semantic zoom feature allows for more information to be shown by zooming in or less to be shown by zooming out.
- Help users recognize, diagnose, and recover from errors: LOW
  The system lacks any form of error messages.
- Help and documentation: LOW
  The system is developed with the intention of being able to use without documentation, but it was shown through expert evaluation that this is not the case for some actions. The system lacks any form of documentation.

6.5 Discussion

The main issue identified through both the heuristic evaluation and the user testing is the lack of mistake proofing and recovery supported by the wall. A common problem was that it was too easy to change the colour of a card, and participants mistakenly did so. Firstly, this should be addressed by making the colour change gesture mistake proof. Secondly, there was no way to recover from the error and return the card to the previous state unless they remembered what colour the card originally was. This is a major problem that needs to be addressed in the future.

An interesting result is the difference of opinions about the features between the participants. There were only a few features that all participants came to the same conclusion about. The teams that they come from all do things differently and a tool developed to work for them should be as flexible as possible. This means maximum configuration allowed without anything imposed on the user. There is some configuration in the prototype, such as allowing for any amount of columns to be used, but not enough according to the participants. The amount of usage forced on the user was small, for example, providing a feature without imposing a use for it, and letting them decide what they would use it for. This was still an issue when it came to laying out the task cards. One participant felt that they would like ultimate control over where the cards were placed. This suggests that even though there was a small amount forced on the user, it was still too much. Things such as the layout of cards should be configurable to be turned on or off, that way the use of the tool can target a wide range of development teams and conform to how they want to use it, not how the wall wants them to use it.

A paper wall allows any amount of customisation wanted. Trying to move it into a digital wall should try match as much of the level of customisation possible. Commonly requested customisation was allowing for any amount of detail to be put on task cards as wanted, free reign over available colours, editing of columns, editing of tasks, editing of stories, and free rotation of the cards.

All participants stressed the importance of having a backlog and archive along with metrics, even when they were informed of the project scope, it was still something that ranked highly on their critique of the tool. This comes down to the participants being domain experts and current agile practitioners. When evaluating the prototype, they viewed it how they would as a completed tool they were using in industry. This provided excellent feedback on what would be needed to turn the prototype into working a tool for agile development.

The aim of space utilisation was not addressed through a single use case in the user evaluation. The intention is that it would come up throughout user test as the participants use the wall. There are three areas that promoted space utilisation that the user evaluation showed. Participants mentioned that the stacking feature would be good to use on large projects as it makes less important tasks take up less space. The layout of the cards, and the ability to resize columns was said to be a useful feature by the participants because often on a paper wall they run out of space to display cards, and they are obscured by other cards on top of them. This shows that the participants noticed the dynamic sizing of the wall and thought it would be useful in a real project

Chapter 7

Conclusion & Future Work

This chapter presents the main conclusions of the project and describes directions for future work.

7.1 Future Work

Future work on the wall would involve fixing some issues and implementing new features that the user test brought to attention. The main thing is the lack of mistake proofing in the wall, and the ability to undo or redo changes. Implementing better error prevention would also solve issues that the heuristic evaluation described. Specific details that can be worked on are:

- Removing the colour change gesture on cards and replacing the tap gesture with a feature to show detailed information about it.
- Adding a customisable list of colours along with a legend to associate them with
- Adding support for avatars to show ownership
- Implementing undo/redo

The wall is also open to the implementation of metrics to be generated about the cards on it, as requested by test participants. This would be useful to management of projects. Work would involve implementing ways to track cards time on the wall, and generate metrics and charts about them.

Two new components that can be added to the system would be the product backlog and an archive. These are two key components that are needed if a complete system was to be use in industry. They provide the complete life cycle of story cards from creation to storage. The implications are that this tool would become a complete project management tool for agile development.

An aspect of the tool that can be focused on would be use in distributed environments. Adding support for teams to use tables at different locations would add another benefit to the electronic wall. One thing to consider would be how to allow for teams to use the wall at the same time. Some form of marker would need to be visible that shows what each user is doing from each location. It would also need consideration on how to handle gestures on an object that two people are trying to handle at the same time.

7.2 Conclusion

Agile story walls are an integral part of agile development. A traditional story wall misses out on the benefits technology can provide because it is a paper artefact. The aim of this project was to create an electronic story wall targeting the dynamic use of space. Four alternatives for platforms were considered, and a multi-touch application was decided on. A prototype was designed and implemented using MT4j over three iterations. An evaluation using three local domain experts and three international researchers was carried out on the tool. This evaluation showed that space utilisation is a useful feature that paper walls are missing, but better error prevention was needed in the prototype. All participants agreed that multi-touch is a step in the right direction to bridge the gap between paper and electronic story walls. This allows for a dynamic electronic wall to provide features not possible on a paper wall, but maintain some of the situational awareness and interaction of a paper wall.

The contributions of this project are:

- A dynamic layout and interactive visualisation of a story wall.
- A proof of concept prototype implemented on a multi-touch table to preserve a more traditional form of interaction.
- An evaluation of the prototype by local domain experts and international researchers.

If the prototype was to be used in industry, it would need further development to include a backlog, archive, and metric generation. Further development would also provide distributed teams with a story wall tool that supports development across multiple locations.

Bibliography

[1] BANDIT SOFTWARE, L. Leankit kanban. http://leankitkanban.com/.

[2] HIGHSMITH, J., AND COCKBURN, A. Agile software development: the business of innovation. *Computer 34*, 9 (sep 2001), 120 –127.

[3] KELTER, U., MONECKE, M., AND SCHILD, M. Do we need 'agile' software development tools? In *Objects, Components, Architectures, Services, and Applications for a Networked World*, M. Aksit, M. Mezini, and R. Unland, Eds., vol. 2591 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2003, pp. 412–430. 10.1007/3-540-36557-5_29.

[4] KOHARI, N. Agilezen. http://agilezen.com/.

[5] LAUFS, U., RUFF, C., AND ZIBUSCHKA, J. Mt4j - a cross-platform multi-touch development framework. *CoRR abs/1012.0467* (2010).

[6] MORGAN, R., AND MAURER, F. Maseplanner: A card-based distributed planning tool for agile teams. In *Global Software Engineering, 2006. ICGSE '06. International Conference on* (oct. 2006), pp. 132 –138.

[7] NIELSEN, J., AND MOLICH, R. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people* (New York, NY, USA, 1990), CHI '90, ACM, pp. 249–256.

[8] REES, M. J. A feasible user story tool for agile software development? In *Proceedings of the Ninth Asia-Pacific Software Engineering Conference* (Washington, DC, USA, 2002), APSEC '02, IEEE Computer Society, pp. 22–.

[9] SHARP, H., ROBINSON, H., SEGAL, J., AND FURNISS, D. The role of story cards and the wall in xp teams: a distributed cognition perspective. In *Agile Conference, 2006* (july 2006), pp. 11 pp. –75.

[10] WANG, X., GHANAM, Y., PARK, S., AND MAURER, F. Using digital tabletops to support distributed agile planning meetings. In *Agile Processes in Software Engineering and Extreme Programming*, P. Abrahamsson, M. Marchesi, F. Maurer, W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, and C. Szyperski, Eds., vol. 31 of *Lecture Notes in Business Information Processing*. Springer Berlin Heidelberg, 2009, pp. 261–262. 10.1007/978-3-642-01853-4_59.