Understanding Digital Cardwall Usage

Stevenson Gossage, Judith M. Brown, Robert Biddle
School of Computer Science
Carleton University
Ottawa, Canada
Email: stevenson.gossage@carleton.ca, mmjbrown@gmail.com, robert.biddle@carleton.ca

***Abstract*—In Agile software development, key artefacts used to support the process are the User Story (usually recorded on a Storycard) and Story Cardwall (usually a dedicated portion of a wall). These low-fidelity tools work together to help teams stay focused and self-manage their projects. The need to support distributed teams and team members makes the physical Cardwall impractical and teams are therefore migrating towards digital story management tools. We conducted field studies of 8 Agile teams using digital Cardwalls, and performed qualitative data analysis to understand patterns of usages and user needs. We identify issues to address in the design of digital Cardwalls.**

***Keywords*—software process; agile software development; software development tools; cardwalls;**

I. Introduction

Agile methodologies promise increased productivity for software development teams by providing proven strategies, guidance and best practices. At the root of the Agile tool-set is the User Story (usually recorded on a Storycard) and Story Cardwall (usually a dedicated portion of a wall). These low-fidelity tools work together to help teams stay focused and self-manage their projects. The Cardwall helps to foster awareness and encourages transparency by acting as an information radiator [1], where the User Stories are on display and stakeholders can easily track the progress of the development work. Physical Storycards and Cardwalls are very low-tech, and often made using index cards or sticky-notes for the Storycards, and the Cardwall often merely occupies empty space on a wall near the developers. However, the need to support distributed teams and the availability of issue tracking software is leading towards *digital* Cardwalls. We wanted to learn how digital Cardwalls are used in practice by professional Agile teams, to see how they compare to physical Cardwalls, and to understand how their design might potentially be improved. Do do this, we conducted field studies, including interviews and observations of software development teams. We adopted a qualitative approach and used thematic analysis to help understand our data and interpret our findings.

II. Background

A. Physical Cardwalls

A natural question might be to wonder how the necessary level of detail and complexity can be captured on a few cue cards pinned to a wall, and, maybe more importantly, how the physical Cardwall actually helps software development teams to meet their goals and deadlines while producing quality code.

Sharp et al. conducted an in-depth five-year observational study of XP software development teams using Storycards and physical Cardwalls and the results were published in several papers. In particular, one paper [2] presents an investigation of a team's use of Storycards and Cardwalls in terms of physical interactions and social interactions. They looked at the use of both Storycards and Cardwalls from the perspective of notation and applied Green's Cognitive Dimensions framework [3] to understand the importance of each notation and how it supports the intended activities in terms of the different cognitive dimensions. The investigation focused on social context and how the social nature of the process binds and supports the use of Storycards and Cardwalls, and addressed how the social context frames the underlying agreements about how these artefacts are used to ultimately support the goal of producing working software.

Overall, Sharp et al. found that the Storycard notation generally supported the following cognitive dimensions: abstraction, closeness of mapping, low diffuseness, provisionality, and low viscosity. User Stories capture requirements and are therefore an abstraction of them, which also means they are necessarily close to the domain which supports closeness of mapping. Low diffuseness is supported by the stories being written in the language of the user, that they are brief and terse by design because they are only intended to be a reminder for further discussion. The Storycard medium on which the story is presented, i.e., on an index card or sticky note, gives the Storycard a feeling of provisionality. This also supports low viscosity because it encourages the engagement of a Storycard by the users of the Cardwall. The Cardwall generally supported the following dimensions: provisionality, low viscosity and process visibility. It is easy to move cards, change labels and, start new iterations, all of which contributes to the Cardwall's high provisionality and low viscosity. Also, the Cardwall's columns help reveal the underlying process and can be easily understood, which makes the visibility of the process high. The physical Cardwall does not have much support for other dimensions, and, for example, is weak with regard to exposing hidden dependencies, preventing error proneness, allowing progressive evaluation, and avoiding premature commitment.

B. Digital Cardwalls

Large high-resolution displays are now readily available, as is the support for multi-touch capabilities. Leveraging these technologies seems like an obvious place to start when thinking about developing a digital Cardwall. Every day more devices are being produced at a reasonable cost with support for two or more simultaneous touches — a critical feature for the development of truly collaborative tools.

a) Research Prototypes: Custom software for digital Cardwalls has been developed and explored in various research projects.

One example of specific digital Cardwall software is the Agile Planner for Digital Tabletop (APDT) [4]. APDT was designed based on a prototype by Weber et al. [5] which was intended for co-located collaboration on a single touch surface. APDT chose to use this as a starting point, but the researchers wanted to enhance it with support for multi-touch, and the ability to interface their Cardwall with other Agile planning tools. APDT was designed after observing traditional Agile planning meetings, as well as meetings conducted using the earlier Distributed AgilePlanner (DAP) [6]. As the name suggests DAP was designed to support distributed Agile teams in the planning and maintenance of an Agile project through the use of a digital whiteboard and Storycards. DAP had been developed with a traditional single user interface paradigm (one keyboard, one mouse), to enable users to collaborate remotely; it did not support multi-user interactions in a co-located environment. APDT also studied and drew from the literature available on the use of multi-touch tabletops for group collaboration.

b) Trello: Trello [7] was created by Fog Creek Software during the course of our own project. Perhaps more than any other digital tool for story management, Trello captures the simplicity of the traditional physical Cardwall. The simple design allows flexibility in how it can be used. Trello can align with many different workflows, from simple to-do lists to Agile development, and also to other personal, business or management applications.

Each new board starts with three empty lists titled: Todo, Doing and Done, but you can add as many lists as you want, each with its own title. Using a card metaphor, in Trello you add new content to a list by clicking the "add a card" button at the bottom of any list. The Storcards have two views, a minimal view used while viewing the board at large and a detailed view, where you can see and edit all the extra content that is hidden on the back of the card. Each list can grow arbitrarily and the Storycards allow the user to easily add rich content, such as images and URLs or even embedded videos. With Trello, everything is saved automatically so there is no need to remember to save or update. Trello has been designed very well from a user interaction perspective, but it does have limitations. For example, it is not possible to view the details of a Storycard while still viewing the Cardwall. Similarly, one can only see the details of one particular story which makes it difficult when planning and the discussion involves the details of more then one story. Finally, Trello is not designed to support simultaneous, co-located multi-user interaction which may have an impact on its support for collaboration.

c) Story Repositories: In addition to software specifically designed for digital Cardwalls, another important influence comes from digital story repositories: for example, JIRA [8] is an issue tracker developed by Atlassian. JIRA allows software development teams to track and assign issues as well as to track the activity of teams and their members. Issue trackers, also known as bug trackers, ticket support systems, or management workflow systems, allow teams to enter and track the progress of whatever the particular system is designed to track. They provide useful search features and reporting

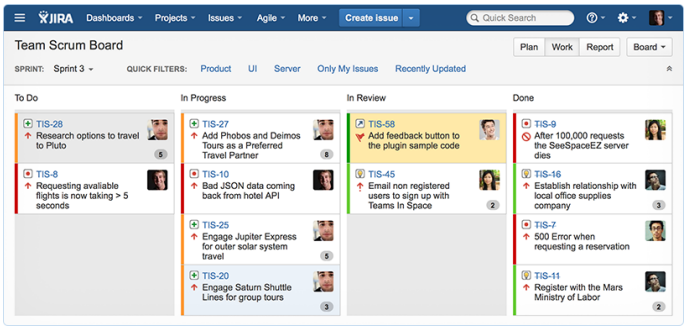

Fig. 1. A typical JIRA/Grasshopper Cardwall used by teams in our study. Each card is associated with an issue in JIRA. These Cardwalls were projected on walls in team meetings. Source: [10]

capabilities, including graphs to help visualize progress, and reports for management.

Such systems are not really designed to support Agile planning with User Stories, but are widely used for this purpose in practice, JIRA being perhaps the most popular choice [9]. The major impetus for this development was that Agile teams were looking for a solution to the distributed team dilemma where team members working from different locations had no access to the team Cardwall. The use of JIRA by Agile teams for this purpose became so popular that a plugin called Greenhopper was developed to add a Cardwall view on top of JIRA. Development for this plugin is on-going and the name has been changed to JIRA Agile; see Figure 1. Several of the teams involved in our field study used JIRA in combination with the Greenhopper/JIRA-Agile plugin.

III. Study

We aimed to understand the reasons leading to the adoption of digital story Cardwalls and the benefits and frustations experienced by teams using digital Cardwalls for their planning or evaluation meetings. In particular, we were interested in answering the following research questions. Q1: In what ways do current implementations of digital Cardwalls and stories meet the needs of distributed Agile teams? Q2: What usability issues exists with the use of digital Cardwalls and stories? Q3: Can we identify features that are difficult, inadequate or missing from existing digital Cardwalls and story implementations?

We decided to conduct a week-long field study that would consist primarily of observations that would capture team behaviours. Beforehand, we conducted a pilot study of a professional team using a physical Cardwall, to atune ourselves to differences we might see in the use of digital systems. In the main field study, in addition to the observed interactions and motivations, we also decided to interview, whenever possible, key team members from our observational study: the project manager, one novice team member and one senior team member. These two methods would help us to develop contextual understanding, insight, and to collect personal opinions about the use of story Cardwalls. We received approval from the Carleton University Research Ethics Board. When interacting with team members and handling data we followed standard ethics protocols; when reporting results we use pseudonyms to mask the names of projects, organizations and individuals.

TABLE I. INTERVIEWEE TABLE

Name	Role	Experience Level	Org.
Mitchel	Software craftsman	Inexperienced	Octagon
John	Developer	Experienced	CareCo
Wayne	Software craftsman	Experienced	Octagon
Michael	Project Manager	Experienced	CareCo
Dennis	Project Manager/CEO	Experienced	Rand
Keith	Software craftsman	Inexperienced	Octagon
Donnie	Project Manager	Experienced	Octagon
Jack	Project Manager	Experienced	Pilot Study
Total: 8			

A. Participants

The target participants of the observational study were established professional Agile teams that used digital story Cardwalls. The participants (Agile Teams) of this study were selected opportunistically, mostly through connections made while attending an Agile conference. Our field work spanned two cities, one in Canada and the other in the United States of America. Including a pilot study (included because it was instrumental in confirming our understandings of physical Cardwall use), we observed at four different organizations/employers, as shown in Table II. At two organizations, we studied multiple teams. There were sixty-four individuals in our study. From these individuals, we opportunistically selected seven participants for in-depth private interviews. The selection was based on their role, availability, and willingness to participate. The demographic data for the interviewed participants can be seen in Table I, which lists the participant's role, experience and employer organization. Two participants worked only on one project, but others worked on two more more.

B. Procedure

We planned to observe professional Agile software teams as they collaborated around Cardwalls using standard ethnographic methods [11]–[13]. We were interested in observing daily stand-ups, Iteration Planning Meetings (IPMs) and retrospectives. The observations were to be conducted in the place of work of the different teams to maintain high ecological validity.

We conducted the interviews in an available office in the place of work of the participants using standard semi-structured interview methods [11], [14]–[16].The private, informal, semi-structured nature of the interviews allowed us to explore the personal experiences with Cardwalls and stories of Cardwall use from each participant. We also aimed to understand our participants' overall impressions about the tools they used, including points of frustration and their insights on how things could be better.

We conducted our observations in the regular workplaces of our participants, and involved materials and equipment that were an integral part of their activities.

C. Analysis

The observational and interview data were to be collected, summarized and typed up (in the case of the observational notes) or transcribed (in the case of audio recordings). The typed notes and transcribed audio recordings would be imported into ATLAS.ti, a tool designed to support qualitative research in the data-analysis phase.

We decided to use thematic analysis [17], [18] to analyse our data; this would allow us to identify common themes across the interviews and observations through the process of coding. To code, we would highlight interesting bits of information and tag them with an appropriate code/label, such as "Cardwall Enhancements", or "Story Grouping." The process of coding is manual and laborious, requiring a significant amount of attention to the identification of significant details. Once data is coded, ATLAS.ti has a number of features which can help researchers identify 'saturated', i.e., rich codes. These include the frequency of a code, i.e., the number of quotes identified and tagged with a particular code. We planned to identify clusters of saturated codes to identify themes, i.e., topics that were commonly raised during interviews or incidents commonly observed across projects.

IV. RESULTS

In this section we present results from our analysis of the observational and interview data and speculate in general terms about the design consequences of these results.

A. Overview

All of the teams participating in our study used Agile processes for development work, typically involving elements from both Scrum and XP (Extreme Programming), and the language they used (e.g. sprint vs. iteration) varied slightly from team to team. Two of the organizations were closely connected and had very similar Agile practices (Octagon and CareCo); these included: pair-programming, daily standup meetings, sprint planning meetings and retrospectives. These two organizations also practised Software Craftsmanship [19], which, although Agile-like in many respects, distinguishes itself through its strong emphasis on mentoring for the purposes of accelerated skills development and general understanding, and a constant exposure to best practices through case studies, workshops, invited speakers and the like.

Our participants were organized into eight teams, each one responsible to a client organization (See Table II). The majority of client organizations were based in the USA and were introduced to us through Octagon. Octagon is a development firm which mainly does contract development work. For these contracts they would either establish a predominantly in-house team to do 100% of the development work, or they would provide developers and sometimes project leads to work directly with the client's own development teams. Either way, the teams were always composed of employees from both Octagon and the client organization. For example, CareCo (an organization we visited for two days to observe and conduct interviews) had contracted Octagon to help two distinct in-house development teams with their work on two different projects. For both these teams, out of seven total members, two members per team were employed by Octagon. Four other Octagon clients created teams that were mostly composed of Octagon employees. With these teams, most of the work was conducted at the Octagon head office; Table II shows the client organizations, team compositions and team sizes.

DemographicInfo DemographicInfo

In all, we observed teams at work on eight different projects. These projects used one of three following tools

TABLE II. TEAM TABLE

Org.	Team's Composition	Team's Project	Size	Cardwall
CareCo	5 client, 2 Octagon	Babysitter WebApp	7	JIRA
Octagon	5 client, 2 Octagon	Nanny WebApp	7	JIRA
Octagon	3 client, 7 Octagon	Online Insurance Claims	10	SBoard
Octagon	2 client, 6 Octagon	Athletics Camp	8	SBoard
Octagon	2 client, 6 Octagon	Online Survey Creation	8	SBoard
Octagon	2 client, 5 Octagon	Event Scheduling WebApp	7	SBoard
Pilot Study	7 client	Auditing Software	7	Physical
Rand	10 client	WebApp	10	JIRA
Total			**64**	

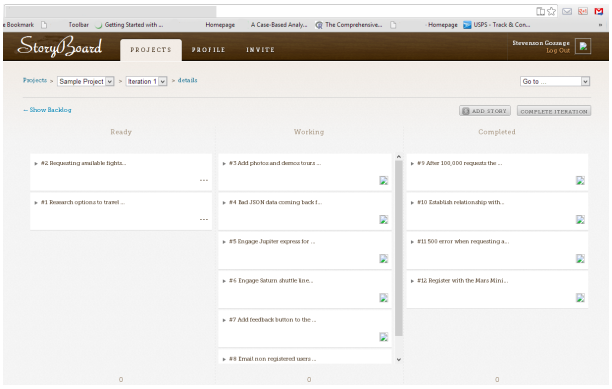

Fig. 2. An example of the StoryBoard Digital Cardwall. There are 3 fixed columns: Ready, Working and Completed. This layout was typical of observed Cardwalls.

as their Cardwall: JIRA/Greenhopper digital Cardwalls, an in-house digital Cardwall called StoryBoard, and a physical Cardwall created with sticky-notes and tape. See Table II for details on the projects and type of Cardwall, and Figures 1 & 2 for Cardwalls like the ones that we observed. All teams used Scrum, though several also included some elements of XP.

We observed 13 meetings where stories and Cardwalls were used, including daily stand-up meetings, iteration planning meetings (IPM), and sprint retrospective meetings. Table III provides key information about the types of meeting we observed across teams. We observed real work-flows around Cardwalls, specific behaviours with respect to Cardwalls, and problems that arose due to the type of Cardwall in use.

TABLE III. OBSERVATION TABLE

Events	Org.	Location	Project
Daily Standup	CareCo	USA	Babysitter WebApp
Daily Standup	Octagon	USA	Nanny WebApp
Daily Standup	Rand	Canada	WebApp
Iteration Planning	CareCo	USA	Babysitter WebApp
Iteration Planning	Octagon	USA	Nanny WebApp
Iteration Planning	Octagon	USA	Online Insurance Claims
Iteration Planning	Pilot Study	Canada	Auditing Software
Iteration Planning	Octagon	USA	Athletics Camp
Iteration Planning	Octagon	USA	Online Survey Creation
Iteration Planning	Octagon	USA	Event Scheduling WebApp
Retrospective	CareCo	USA	Babysitter WebApp
Retrospective	Octagon	USA	Nanny WebApp
Retrospective	Octagon	USA	Online Insurance Claims
Total Events: 13			

Interview participants varied substantially with respect to their role on the team. Two were novice developers and six were experienced (including three project managers) (See Table I). Experienced interviewees had more than five years experience in software design and development, a figure which correlated closely with their experience with Cardwalls and stories. Inexperienced participants had five years or less experience on software teams. All participants had experience on previous projects because most of the current projects were newly established. Some were working on only one current project, several on two, and one worked on three projects. Interviews ranged from 34 to 55 minutes.

We asked interviewees about their experiences with Cardwalls, both current and past. We asked questions about behaviours we had seen while observing Cardwall use in the pilot study and asked 'why' questions about their particular preferences and tendencies when using stories and Cardwalls. In this way we were better able to understand the behaviours we had observed and also the historical and organizational context of Cardwall usage.

In all we developed 57 different codes, however, 10 of these were not saturated [20] so we eliminated these from our analysis, leaving 47 codes which we grouped into 6 different themes. Table IV shows groups of saturated codes associated with their corresponding theme. The following sub-sections describe 5 of the 6 themes that we identified based on the application of thematic analysis to our data (we don't report on the minor theme Keeping the Cardwall Updated). The content of all themes are based on our notes of team meetings and our interview transcripts. Throughout, we used quintessential participant quotations that effectively express the message we received from them as a whole.

TABLE IV. CODES BY THEMES

Themes	Codes
Flexibility Advantages with Digital Cardwalls	big picture, distributed, resources, simple, Cardwall_digital, Cardwall_enhancements, Cardwall_location, Cardwall_meetings, Cardwall_backlogs, Cardwall_planning, Cardwall_process, physical vs digital, Negatives, Positives
Challenges in Presenting Detailed Information	Negatives, Positives, physical vs digital
Exploring and Filtering Information	story_acceptance criteria, story_alignment, story_annotating, story_components, story_grouping, story_traceability
Managing Backlogs	story_activities, story_creation, story_estimation, story_expectations, story_prioritization, software delivery, sprints
Multi-disciplinary use of Stories	roles_Cardwall user, roles_customer, roles_designer roles_developer, roles_end user, roles_product owner, roles_project manager, roles_QA, roles_stakeholder, roles_various
Keeping the Cardwall Updated	foster_awareness, foster_comunication, Cardwall_visibility, collaboration, story_views, story_testing, story_states, story_ownership

B. *Flexibility Advantages with Digital Cardwalls Theme*

This first theme is important but quite general, and many ideas in this theme also emerge in following themes. Part of our data for this theme comes from our observations of Cardwall use. In addition, the interview transcripts captured many negative and positive views about both digital and physical Cardwalls. In general the advantages and disadvantages covered in this theme all touch on the idea of flexibility and how the digital Cardwall provides more ways to do things, but also how this can over-complicate the tool and impact its usability. The remainder of this section organises the perceived advantages and disadvantages using the following sub-themes:

wall and user location, *complexity*, *relationship visualisation and traversal*.

1) Wall and User Locations: The teams we observed mostly used projectors to display their Cardwalls. This was especially true of the Cardwalls used for external customer projects (most of the projects), but not so true for Cardwalls that were created for internal projects. Projected Cardwalls have the advantage that they are portable and can be used in any available meeting space where there is a projector. When used in this way, we observed that one member of the team (usually the project lead) was responsible for updating the Cardwall as the team worked.

Keith, a novice developer had used a physical Cardwall on a previous project. Keith liked the physical presence of the Cardwall and that it afforded reflecting on the big picture.

> "... with the physical [Cardwall], its just a lot more visible, I guess. ...the virtual one, the online one, is easily accessible [for updating purposes] ...[but] the [physical] one in the workplace is just easier, ...you can sit back, and ...have a full view of what's in progress" (Keith).

Another disadvantage of physical Cardwalls is the poor fit between physical Cardwalls and the increasingly common occurence of distributed team work. We observed one or more team members located off-site, working from home or travelling. In fact, there was at least one distributed (remote) participant at each of the iteration planning meetings as well as in some of the daily stand-up meetings. Remote users communicated via Skype, GotoMeeting, and teams even used telephones when the other two technologies were not co-operating (something we witnessed in two meetings).

A disadvantage of physical Cardwalls is the limited physical space teams have for display purposes. In the environments we observed, there were many small projects, and they shared common meeting rooms for their team meetings/team events. In fact, the requirement for wall space outgrew the physical resource. Further, because customers attended these meetings it was not appropriate to display Cardwalls from other projects; the projector system allowed the team to have only one Cardwall visible at a time, and made efficient use of wall space.

A minor point of frustration that was expressed for digital Cardwall use was the necessity for extra steps to make them work. Michael expressed this point, but also indicated that because of the distributed nature of their teams a digital Cardwall was a necessity:

> " ...in general I do like the Cardwall we have[JIRA/Greenhopper]. I've used just physical Cardwalls and they're more effective in terms of being able to rapidly capture thoughts and easily move things around. ...That said, we work remotely with developers and need remote resources so a physical Cardwall just isn't possible in our situation."

2) Complexity: Wayne speculated about problems with maintaining either physical or digital Cardwalls, especially when projects were large. For digital Cardwalls, he saw the problem of page refreshing as an annoyance, but with physical Cardwalls he saw clutter being a challenge because of space limitations.

> "If there were maybe a larger team with a longer sprint, there would be so many cards that it might help to physically rearrange them [to reduce clutter]. But we're not in that situation."

Michael, talked about how he would love to see even higher level overviews of the project, and not just the current iteration. He explained:

> "I think one of the biggest shortcomings ...with our Cardwall solution is, within an iteration it works pretty well, but if we want to zoom out and see a story map of the entire backlog, there's not a view available. So, I think, in terms of understanding the larger context and being able to visualize that, I think that would be a great benefit for everyone involved in the project. But it's not something that we can do with our current tool. [JIRA/Greenhopper]"

Interestingly, the added benefits afforded by the digital environment can be viewed as either a blessing or a deterrent. Deterrents that can have a negative impact on the team's willingness to use a feature. For example, Michael, one of our project managers, loved how you could link JIRA with other tools and described this feature as important and easy to use. Michael said,

> "One of the things that JIRA does really well is when developers commit code, the first part of the comment is the key of the issue in JIRA, and as they commit, there's an integration between GitHub, which is our source code management and JIRA. So, Commits are linked to JIRA issues. And then within JIRA you see all of the activity related to that particular issue. So I think that simple linking mechanism has 'incentivized' people to use it more. Because now the team knows we have a complete traceability from ...the requirements, all of the information about the requirements, the status, the captured history, and then a link to the actual changes in source code."

At the same time Wayne, one of the developers working with Michael recounted his frustrations relating to the same issue of code change notifications, he said:

> "I get emails when stuff changes ...it's like, [if] my updates happening 10 times, I get 10 emails", which "is completely meaningless to me because there's too much information."

Wayne recounted, "My inbox is sacred. I don't want to use email to notify me that stuff is happening. I feel it's the wrong medium." He did not want to be flooded with meaningless code change emails. We concluded from this and other similar stories that it is extremely important to consider *how* changes and relationships, dependencies etc. can be visualized so that we accomplish the goal of increasing team and individual awareness without creating unnecessary distractions and annoyances.

3) Relationship Visualisation and Traversal: In our data, participants mostly described how digital stories enhanced their work and had the potential to enable it further. Wayne used JIRA to track stories and said stories acted as a helpful guide for his development work. "All I need is a few sentences telling me what to do, and a name for those few sentences." He sees stories as "buckets of criteria [requirements]", but would like to easily see relationships between stories.

In the physical Cardwall we observed in our pilot study, the visualization of relationships between stories was supported through the use of simple methods like their use of colors,

annotations, and rows. The rows (commonly referred to as swimlanes) enabled the team to visualize the flow of stories that had been been grouped together based on their common relationship to a particular role (in this case the relationship was 'owned by'). In general, swimlanes are used to visualize the flow of related stories, where the stories from each row can be worked on independently from the stories in other rows. Swimlanes can be based on virtually any relationship useful to the team and examples may include the role of the team member, the role of a sub-team, a software component or feature that the stories are part of. In the electronic Cardwalls we observed, we saw support for some of these same simple methods, but we did not see teams attempt to visualize any sophisticated types of story relationships. We later learned from the interviews that the ability to see these relations was either difficult to do, or not supported by the tool. When discussing how a digital Cardwall could be used to visualize different types of relationships, Michael said:

> "I think the key to success there is the linking mechanism. In JIRA they do have the concept of a link. But you have to click more actions and then link, and then you have to pick the type of link, and then you have to search, and then you have to try to find it, and then you click it, and then you click another thing, and it's [just] too much. It's way too hard to do, so nobody does it."

Dennis and his team used JIRA and would have liked to see relationships between stories made more evident. When they worked, Dennis explained that his team kept track of dependencies between cards "in their heads" because not only could it *not* be done via the Cardwall, but in general "we don't have good dependency tracking tools." Dennis also believed it was useful to look at the evolution or history of stories because he wanted to see a hierarchical understanding of stories and how they evolved from large epics to smaller implementable stories. "I think it's very, very important to be able to look at the evolution." Dennis' interests then were in linking dependent stories and, like Wayne, linking stories in a hierarchy from epic stories on down to the type of stories seen on a typical Cardwall, i.e., stories that can be completed within one iteration. However, Dennis was frustrated with performance issues in general, "the performance of the tools is just terrible! I mean humans are very fast relative to most software," and he claimed that any tool he would want to use would have to perform better than the tools they were currently using.

Related to the previous issue was the frustration of not being able to navigate quickly between an epic and a corresponding story and vice versa. John, one of the more experienced developers, recognized the benefits of being able to decompose and reconstruct epics into smaller, more manageable stories, but also thought it was important to be able to leverage these relationships as a way to quickly jump from one story to another,

> "Ideally, if you can split an epic [into multiple stories] and keep that tied back to the epic ... that would be really nice to be able to just go-to-it [the story or epic] depending on which direction you wanted to go".

C. Challenges in Presenting Detailed Information Theme

Large higher resolution displays are another way to display more content; but the necessary physical size of Cardwall elements may be a more important driving force towards favouring larger, rather than higher resolution displays. Extremely high resolution may be more important for personal displays and portable devices as high resolution text is easier to read and causes less eye strain. Small touch-enabled portable devices like smartphones may have extremely high resolution, but the user experience is of utmost importance and the designers of these systems balance the added higher resolution with the knowledge that the interactive components and widgets of the applications must still be a size that is sufficiently large so that the fat finger problem is avoided. This same issue also applies to text, the added resolution makes a font-size appear much smaller than on a device with less resolution; the font can be too small though and application designers must find a sweet spot that works for the majority of the target audience. Similarly, collaborative applications specifically designed for either high-resolution displays or tiled displays must also consider how text and other artefacts will be viewed and at what distance.

For the interactive touch-enabled Cardwall, the addition of a touch interface may further complicate this because, like the fat finger problem found on smartphones, the interactive application components on a high-resolution display must not be rendered so small that they become difficult to use. The trade-off between the size and resolution of displayed objects is key for Cardwalls, and this point was expressed by our participants who were concerned that individual objects (such as Storycards) need to be big enough to be understood, but not so big that a big picture can't also be discerned.

> "So I mean I could easily be biased to use a Cardwall if the cards were fairly small if I could see them, if I had a mirror, if I could slide over the card and [see a balloon view] so I could see the full description so I can press a lot of cards into the screen [itself] you know... I don't think I'm really that biased against [digital] Cardwalls it's just that [with] the current ones the cards are either too big or too small. (Dennis)"

Dennis also raised concerns about inflexibility in the size of physical or digital cards, especially when there were many cards. "If they're ... too big [you can have problems] if you want to actually manipulate a whole lot of cards. And if they're too small, you put them all up there and you can't read them." It was important to Dennis to find the right balance "between the manipulation of many cards or reading a single card" because "you need to do both." When asked about his experience with the JIRA Cardwall, another participant, Keith, replied, "It's OK. It's a little cluttered."

D. Exploring and Filtering Information Theme

The difference between novice and experienced Agile team members is most evident when participants were asked about their perception about how stories and Cardwalls facilitated awareness. For example, one experienced Agile practitioner, Michael, describes how Cardwalls help him keep track of the status and progress of multiple projects, "To get an idea of where things are from a status perspective. So, I have this team, but I also have five other projects that are going on." Wayne, an experienced team member, wanted to visualize the relationships between stories. He envisioned "a robust framework for indicating that this story came out of this other story" and something to indicate "here's what its original form was" and something that would tell him "here's the spawn of that story."

He'd like to see stories linked in various meaningful ways, especially closely-related stories, but also previous versions of a story or even a story's "history." In contrast, less experienced Agile practitioners like Keith would focus more on the physical and visual details of the Cardwall, which Keith felt helped to increase the team's overall awareness. For example, he described the use of colors to indicate the likelihood a story would be completed. In essence, the more experienced team members were more concerned with the big picture relative to the less experienced team member.

Opinions about the importance of grouping, filtering, tagging and linking stories, as well as how, and to what extent, their current electronic Cardwall supported these features was another point where the experience of the team members made an evident difference. The more experienced participants envisioned how added functionality could help increase awareness about important, but currently invisible aspects of their projects. For example, Dennis explained that he would like to see the software move beyond simple grouping and allow him to retrieve vital information about a certain grouping of stories, he says he would like to be able to ask the Cardwall questions like: "how long will this [group of stories] take,..., how many defects are [associated with this group], who's working on [this group], [and] are we behind?" the less experienced team members focused more on the rudimentary ways they currently were able to group stories like Keith who mentioned how they would use colour to identify bugs and tests: "tests were yellow, and bugs were (pink or red)."

Dennis suggests that one useful method of grouping stories would be by software component, so that the team could select different components and see the associated stories, the progress of those stories etc.,

> "We'd like to be able to put the stories in different places [at the same time] so we can [also] see [other issues of importance like] who's impacted and so on, and again, most systems don't allow you to, for instance, ... tag the [stories] by what component [the story is associated with]."

More experienced team members, like Michael, recognized how the Cardwall and the use of stories were actually themselves catalysts for many of those same conversations,

> "I think that people need to see that overall context and need to be able to go up to the Cardwall and have conversations around it, because, if you're just in the weeds, you lose the [big] picture."

By comparison, less experienced team members also found the Cardwall useful for keeping them up to speed on the progress of other team members, "We usually meet around the Cardwall. You'd take a look at what's up and what we have started or finished," but they were not necessarily convinced that the Cardwall was the best medium to help increase their individual awareness about the progress of their team's efforts. In fact, Keith says that he feels he can achieve similar levels of awareness solely through the ongoing conversations between team members, "Usually everyone was talking and [therefore] I knew what everyone's doing..."

Michael also described features that he would like to see available in his digital Cardwall (he is using a JIRA Cardwall) that would help him remain aware of business goals. He said,

> "It might make sense to do things differently from a technical perspective. And that process of reworking the stories is painful. So I don't know exactly the solution, but it would be great to be able to rapidly create new things that represent the technical, and tie those together with the business requirements."

As a manager, he recognized the importance of being able to easily drill down from higher-level, more abstract ideas to lower-level, more detailed oriented issues. For example he felt that the ability to visualize high-level features on a Cardwall would be ideal for meetings with high-level executives; further the ability to drill down from there and see the epics that would be associated with each feature, and in turn see how each epic was broken down into stories, would help increase the overall awareness of the projects' real progress and help provide a more realistic understanding of the work involved in the implementation of a feature. In contrast, inexperienced developers did not discuss business goals.

The opinions of our participants about the value of Cardwalls and user stories differed based on the years of experience with software design and development. In general, we observed that experienced team members envisioned more sophisticated, yet simple features that would help increase the visibility of connections between stories, their history, how they relate to higher level requirements, components or even business objectives and goals, all of which could be used to justify the existence and even the prioritization of stories. Less experienced team members were generally less likely to recognize the importance of such features because they tended to be less focused on the big picture and more concerned with the day to day progress of the current sprint.

Dennis complained about using Greenhopper with JIRA when working on large projects with huge backlogs. "Rand does not [generally] use Greenhopper. There's too many stories and too many cards. You have to have a useful UI rendering. [In JIRA we] can't allow ranges for dates or hierarchies of stories. It's also slow. Some people do use it though." Dennis continues to explain that the underlying databases used by systems such as JIRA do not have a useful data model for the purposes of searching and filtering data relevant to stories and iterations. He says, "Well, the problem is they're fast but, again, you're representing hierarchical information and you need a different data model — most of these things are really designed about, you know, basically storing a story as a key." Issue trackers like JIRA, which have been re-purposed by Agile teams to manage their stories are not able to efficiently search and filter since often stories are stored as blobdata using a single key; queries are extremely inefficient since key terms of interest to Agile teams and key components from story artefacts are not indexed or directly searchable by the underlying database.

E. Managing Backlogs Theme

Opinions of the significance and management of backlogs differed based on the size of the project the team was working on. The backlog is a significant element for large projects; the ability to see the backlog's stories, and to sort and prioritize the stories becomes increasingly important as the size of the project increased. However, larger projects tended to have larger backlogs, which themselves can be displayed

on a Cardwall, as happened with several of the project teams that used StoryBoard and JIRA/Greenhopper, both of which allowed developers to drag stories from their backlog onto a Cardwall. This evolution of the use of traditional Cardwalls is an ideal example of how a digital implementation can enhance the traditional physical Cardwall.

The interviews helped us understand how the teams were truly self organizing, and helped us understand the process that teams followed for choosing stories. Team members were very interested in the prioritization of the backlog and how the stories were selected for each new iteration. We learned how stories were chosen by individuals during an iteration, as well as how teams managed the exceptional case when a developer was left without a story to work on.

Stories were not assigned; each team member would simply pick a story from the ready column and start working on it.

> "So, if there's something in progress, and a person wants to pair, I'll do that. Like, I'll just ask .... And if he doesn't, I [just] pull the [story] out of the Ready column. As soon as I finished that [story], I'm gonna help my other teammate on a story that's in progress. ... its not complicated. (Wayne)"

Wayne continued to explain that noone forces a story on anyone and that he liked this because he felt it meant you would work on parts that you might not otherwise get to see. He said, "Yeah, it's really good to have responsibility and to get exposure to parts of the app which I might not touch otherwise, if I was specialized in one component." Wayne explained that the stories in the current sprint all shared a similar priority and that it was before the start of an iteration, during its planning (the IPM), when prioritization was critical. When asked about the priority of the stories ready for development Wayne replied:

> "No, I think that once we commit, we should try and get them all done. And if it's a one week sprint it's not terribly high overhead. Just do it all, and if one [story] came out [of the backlog], it was as important as the other [that came out of the backlog]. For a one-month sprint maybe there's a disparity in the priority. But for one-week, if we're actually doing the highest priority stories, there's not much difference between the top and the bottom. The top prioritization is [done] in IPM."

Mitchel explained how the prioritization would actually occur in the backlog:

> "So there's the concept of the In Progress list, and those are not sorted by priority. Those are just a bucket of things to be done. The backlog is the only thing that's really sorted by priority. So [in] the backlog, whatever is first is most important, that goes into the [In Progress] bucket first."

If a developer was done and no one wanted to pair, then his next story would come from the backlog. Keith explained how his team decided to use colours during IPMs to identify extra stories not included in the iteration, but that could serve as a mini-backlog. Keith said:

> "We'd usually get the iteration done, and if we didn't then we'd — or if we were done super-early we would grab from the backlog anyway, ... Yeah, we would ask [the product owner], it's like, we had, we ended up having like a red, yellow or a green, yellow, red sort-of-section [on the Cardwall]. So, green: what we are pretty sure that we're gonna get done this week. Yellow: we think we still can [do it], but it's possible it won't get done. And then, red [we might do this, but probably not]. So we had enough [stories] so that we should have extras."

Dennis' team was generally less concerned with the priority of stories, but as deadlines approach and a large release is impending, he admits that, "When we're down to the end of the final time boxes for the release, we're gonna be a lot more concerned about prioritizing what's on the backlog."

In our study we saw how teams sometimes used different backlogs for noting infrastructure-type work that needed to be done, but that did not come from the stories created by the customer. Keith explained how the team started this practice after a retrospective, which identified a need for tracking the creation of lower-level components. He explained:

> "We ended up creating a Dev backlog for stories that developers wanted to do. Like clean up something, or look into some new technology or something else. We put that on a separate wall nearby. [It] just had things that we wanted to do, and [showed] whether you're working on it. You would just remove it if it was done."

Sometimes, teams wanted to use the backlog to store information that would be useful for planning, estimating and reporting purposes. Dennis explained:

> "Bugs are also put into the backlog. Vacations are also put into it. This lets managers understand what is going on if the velocity is reduced or if they're on vacation and training. Everything is thrown into the backlog. This makes for visibility."

Dennis continues to explain how this is important for overall team awareness and planning.

> "If I'm gonna be completely lean and agile, I want to use a single mechanism for doing that. So we like to throw everything in [the Backlog], or encourage people to [do it]; [But,] not everybody does this. [If we put absentee information in the backlog] then I can see there are really only three people there 'cause everyone else is off on vacation, otherwise we'd get bogus reporting systems and metric systems that trigger and say 'look the velocity of [this] teammate's gone down, and so on, and then you go in and find out it's because people are working on three projects, or they're on vacation or training or something. So we like to have complete backlogs [with many types of information]."

F. Multi-disciplinary use of Stories Theme

The interviews made it abundantly clear that stories are vital to the designers and project managers and that they enjoyed and valued working with them. Digital stories in particular were a key artefact for all the participants.

In our study, the process surrounding the use of Cardwalls was basically the same across teams. Teams tended to use basic Cardwalls comprised of three columns: To Do, In Progress and Done. However, some teams extended this simple arrangement with the addition of extra columns thereby increasing the number of states and providing those teams with a better representation of their workflow and their more sophisticated processes.

In most meetings, we also observed that individuals in roles other than the developer role also interacted with Cardwalls through stories. Although initially designed for the developer

community, Cardwalls are becoming a multi-disciplinary team tool. For example, we observed designers and testers organizing their work around stories. However, at the same time we also noticed that non-developers had different requirements for Cardwalls.

We noticed that the effectiveness of the information radiator aspect of a wall depended on the role of the team members using the wall. For example, for developers and designers, the constant display of the current sprint is very useful for increasing the awareness of the status of the project, and it can even serve a motivational function. However, for the team's testers, their interest in the project is probably limited to the list of stories that require testing. For example, if the wall has a testing column, the tester may only need to have access to that part of the developers' Cardwall as a single column to develop an awareness of the size of their task. This suggests that filtering of Cardwalls would be a useful operation. However, for testers, it would also be important to understand the role(s) each story has in terms of the overall system under development. Further, for proper regression testing, and some other more complete whole-system testing it would be to their benefit to be able to visualize which components are touched by a story. This could help identify the interfaces that need testing.

One new possible feature that was discussed was different ways of linking external, but related, artefacts. For example, Michael describes how he would like to link their automated acceptance tests with their Cardwall. They use an automated testing suite called Cucumber, where each cucumber represents an acceptance test, Michael explains the process:

> "So initially [the] Product [group] is writing the Cucumbers [i.e., the tests]. There's a team collaboration to do refinements. Once work is committed then developers are actually linking the Cucumbers into the code and making them pass. QA is also collaborating with developers to make sure that it's done properly, and owns the overall organization of the entire test suite. So there is some manipulation that happens after Product originally writes the Cucumbers. But that's all done between Dev and QA, and Product's out of the loop at that point because it's within GitHub. So we haven't found a good way to maintain that collaboration. And for me that would be through JIRA, since everybody's using that tool, and understands that tool, and is used to collaborating there."

V. Research Questions Answered

A. Distributed Agile Teams

Our first research question was, "Do current implementations of digital Cardwalls and stories meet the needs of distributed Agile teams?" As projects increase in size, the need for supporting distributed team members collaborating around a Cardwall is increasing and the need is becoming unavoidable. Current digital Cardwall solutions support distributed teams in that their systems generally allow multiple simultaneous access. Remote users can login and see the same view as their co-located team members. In general this strategy requires that local Cardwall updates be saved to the server or cloud and sometimes remote clients may even require a refresh to force synchronization between clients so that changes can be seen. In addition, there is no digital Cardwall solution that supports all aspects of a collaborative distributed Agile team meeting using the Cardwall as the central artefact. To support this type of team meeting, the system might benefit from integrated support for both voice and video, although the meetings we observed always used some other software like Skype for voice and/or video communication.

While distributed team members can now login and see the same view, it is not clear how effective current solutions are at drawing the user's attention such that changes are easily seen even by remote participants. Creating this type of team awareness, where the attention of all participants can be drawn to notice changes is critically important and remains an open opportunity for further investigation and research.

B. Usability of Digital Cardwalls

Our second research question was "What usability issues exist with the use of digital Cardwalls and stories?" It was clear that current digital solutions were not necessarily easy to use, personalize or adapt to a teams' specific needs. For example, filtering was difficult, in fact, so much so, that we did not observe a single occurrence of it; the issue only came out during the one-on-one interviews. Other issues, like the lack of control over the granularity of email alerts, where users would complain that when they are bombarded with updates the value is lost and eventually the 'update' emails are simply ignored. It was apparent that updating and remembering to save changes was also an issue (although we did not discuss this theme in this paper due to space constraints). Usability is one of the most important aspects that directly correlates with product adoption. This is why these usability issues need to be understood and addressed in the design of our Cardwall.

C. New or Improved Features

Our third research question was "Can we identify features that are difficult, inadequate or missing from existing digital Cardwalls and story implementations?" Several opportunities for new features were identified. These could have real implications on awareness, not just at the project or iteration level, but also at a third higher organizational level. Among these features, the ability to easily visualize hierarchical relationships and dependencies between stories appears to be critical in terms of fostering awareness at different levels of the organization. Also important and related to the first two levels is the ability to decompose epics into stories and reconstruct those epics from the stories. Another request was the ability to easily link or create associations between stories, and an ability to see their history (including the reasons for the stories and why a story may have changed or been split). Linking business goals and software components was also a requested feature that was currently unsupported.

VI. Conclusions

In this paper we reported on our study of digital Cardwalls for Agile software development. Our field study involved the observation of 8 Agile teams using digital Cardwalls, involving 64 participants. We used a qualitative thematic analysis approach to analyse our data and identified 6 emergent themes. Our research questions concerned digital Cardwall practice and how well current solutions support Agile teams;

we also wanted to identify specific opportunities to improve current solutions. We found that the tools being used by the participants in our study allowed the teams to successfully collaborate with distributed team members, and supported useful flexibility.

Our study was qualitative in nature, with a focus on in situ observations and interviews. It offers useful insight, but future work should address a broader range of projects, domains, and tool support. A diverse range of experience would be necessary to establish generality. We believe, however, that our work identifies key issues to be addressed in the future design of digital Cardwall software. As new digital Cardwalls are developed, we feel it would be valuable to study them and their usage considering our guidelines. In particular, we found potential value in the improved management of the 'backlog', in support for multi-disciplinary stories, and in the exploration and awareness of 'relationships', as follows:

1) Backlog: Traditional Cardwalls focus on the current iteration. For example, they feature columns of stories, where the column specifies the state of the story with regard to its implementation progress. However, the process of planning and selecting the stories for a particular iteration is much more about the backlog and its exploration. In our field study we found our participants frequently needing to explore and discuss stories within the backlog. It was important for them to identify relationships between stories to enable the proper prioritization of stories.

2) Multi-disciplinary use of Cardwalls: Our studies showed that it was not only software developers that organized their work around stories. QA testers and UI designers were also strongly involved in the projects, and they wanted to create stories that particularly addressed their needs. We saw a desire for stories to support features relevant to those roles, and also an ability to filter for stories based on roles.

3) Relationships: The need to support the exploration of relationships and the filtering of results was perhaps the most important finding from our study. As the work in this thesis progressed, it became clear that the ability to explore relationships between stories was of critical importance. The work of Sharp et al. [2] used the Cognitive Dimensions evaluation framework to identify strengths and weaknesses. The weaknesses included hidden dependencies, meaning that the relationships between stories was not available. Surprisingly, therefore, we found that a key weakness in digital Cardwalls was the same as that in physical Cardwalls. The difference is that in digital Cardwalls, unlike physical Cardwalls, this can be addressed by designing software to represent the relationships, and to leverage that in interaction design through interactive visualizations. Moreover, the results from Sharp et al. suggest other weakness that may stem from hidden dependencies, such as error proneness, premature commitment, viscosity, and hard mental operations. Strong support for relationships in digital Cardwalls may also help address these shortcomings.

ACKNOWLEDGMENT

We thank the participants in our study, who openly welcomed us and shared their time, experience, and knowledge. We also thank the Natural Sciences and Engineering Research Council of Canada for funding through the SurfNet Strategic Research Network.

REFERENCES

[1] A. Cockburn, *Agile Software Development: The Cooperative Game, Second Edition.* Addison-Wesley Professional, 2006.

[2] H. Sharp, H. Robinson, and M. Petre, "The role of physical artefacts in agile software development," *The Interdisciplinary Journal of Human-Computer Interaction*, vol. 21, 2009, issues 1-2.

[3] T. Green and M. Petre, "Usability analysis of visual programming environments: A 'cognitive dimensions' framework," *Journal of Visual Languages and Computing*, vol. 7, no. 2, pp. 131 – 174, 1996.

[4] X. Wang, "Using digital tabletops to support agile project planning." Master's thesis, University of Calgary, Department of Computer Science, Calgary, Alberta, Canada, September 2009.

[5] S. Weber, Y. Ghanam, X. Wang, and F. Maurer, "Apdt- an agile planning tool for digital tabletops," in *Proceedings 9th International Conference on Agile Processes and eXtreme Programming in Software Engineering (XP2008)*, 2008.

[6] R. Morgan, "Distributed agile planner: A card-based planning environment for agile teams," Master's thesis, University of Calgary, Department of Computer Science, 2008.

[7] Fog Creek Software, "Trello," Web, Feb 2014, https://trello.com/.

[8] Atlassian, "Jira," Web, Feb 2014, https://www.atlassian.com/software/jira.

[9] F. Chabanois, "Tools used for agile projects: the survey's results," Web, Mar 2014, http://www.devsnotebook.fr/index.php?post/Tools-used-for-agile-projects-%3A-the-survey-results-!

[10] Atlassian, "Greenhopper," Web, Feb 2014, https://confluence.atlassian.com/display/AGILE/GreenHopper+6.0+Release+Notes.

[11] J. Lofland, *Analyzing social settings: a guide to qualitative observation and analysis*, 4th ed. Wadsworth/Thomson Learning, 2006, ch. 5, pp. 81–116.

[12] R. Sanjek, "Fieldnotes: The makings of anthropology," *Journal of the History of the Behavioral Sciences*, vol. 28, no. 3, pp. 274–276, 1990.

[13] H. Bernard, *Research Methods in Anthropology*. AltaMira Press, 2011.

[14] J. A. Smith, *Rethinking methods in psychology*. Thousand Oaks, 1995, ch. 2, pp. 9–26. [Online]. Available: www.summon.com

[15] A. Fontana and J. H. Frey, *Interviewing: The Art of Science*. SAGE Publications, 1994, pp. 361–376.

[16] H. J. Rubin and I. Rubin, *Qualitative interviewing: the art of hearing data*. Sage Publications, 1995, ch. 5, pp. 93–121. [Online]. Available: www.summon.com

[17] J. F. Gubrium and A. Sankar, *Qualitative methods in aging research*. SAGE Publications, 1994, vol. 168, ch. 12, pp. 189–210.

[18] N. Fielding and R. M. Lee, *Computer analysis and qualitative research*. SAGE Publications, 1998, ch. 2, pp. 21–55. [Online]. Available: www.summon.com

[19] Software Craftsmanship Signatories, "Software craftsmanship," Web, Feb 2014, http://manifesto.softwarecraftsmanship.org/.

[20] J. Corbin and A. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, 2008.