

PREPRINT of paper (Please refer to the paper for changes):

Sharp, H., Robinson, H., Segal, J. & Furniss, D. (2006) The Role of Story Cards and the Wall in XP teams: a distributed cognition perspective. To appear at Agile 2006.

The Role of Story Cards and the Wall in XP teams: a distributed cognition perspective

Helen Sharp*, Hugh Robinson*, Judith Segal* & Dominic Furniss**

*Centre for Research in Computing, The Open University, Walton Hall, Milton Keynes MK7
6AA UK

**UCL Interaction Centre, University College London, University College London, Remax
House, 31-32 Alfred Place, London WC1E 7DP, UK

(*{h.c.sharp, h.m.robinson, j.a.segal}@open.ac.uk; d.furniss@cs.ucl.ac.uk*)

Abstract

Much of the knowledge used within an XP team is tacit, i.e. it is hidden and intangible. Two tangible artefacts that carry information about the team's work are the index cards which capture stories and tasks to be implemented and the wall where they are displayed (which we refer to as the 'Wall'). It is widely acknowledged that these are key elements supporting the work of the XP team, but no systematic investigation of their role has been reported to date. In this paper, we focus on the use of these artefacts within one XP team. We use distributed cognition, a framework for analysing collaborative work, to explicate the information flows in, around and within the team that are supported by the index cards and the Wall. We then interrogate the models produced using this analysis to answer 'what if' questions.

1. Introduction

The use of index cards is widespread among XP teams – most often to capture stories or tasks. Earlier observation studies have highlighted the variety of roles that story cards play within XP development, e.g. as progress trackers, as a means of controlling work, as rough paper, and so on [1]. These simple yet powerful artefacts provide key tangible evidence of information sharing within an XP team.

It is recognised within the XP community that one reason this simple approach works well is because of the wider ecological system that supports XP [2]. One element of that wider system in most teams is what we have dubbed 'The Wall' (see Figure 1 for an example). This is where story and task cards for the current iteration are kept, and it is a central focus of

development activity. 'The Wall' forms part of Beck's [3] informative workspace, and is one incarnation of Cockburn's [4] information radiators concept.

However we have found no reports of a systematic analysis that seeks to explicate exactly how story cards facilitate communication and support the main goal of XP, i.e. developing code that provides value to the customer. Nor have we found such reports of the role that 'The Wall' plays.



Figure 1. One example of 'The Wall'

In this paper we focus on one mature XP team and describe and analyse their use of story and task cards, the relationship of these cards to 'The Wall' and the relationship of the cards to the rest of the XP team. We do this using the distributed cognition framework for analysing collaborative work [5], in particular employing the Distributed Cognition for Teamwork

(DiCoT) method [6, 7, 8]. The next section presents our approach to data gathering and analysis, Section 3 presents the analysis itself, and Section 4 presents some discussion of our findings. In Section 5 we outline some future work.

2. Data Gathering and Analysis

Our data gathering and analysis were informed by the DiCoT method. In this section, we briefly describe the context of our fieldwork, then discuss distributed cognition and then DiCoT.

2.1 Observational fieldwork

Two of the authors conducted an observational field study of a mature XP team based in London. The company develops and maintains travel information webpages and travel alerts for a variety of customers in the UK; there were 23 developers in the team, who worked closely with a project manager, and two business development staff (who took the customer role). At the time of the study, the team worked on one three-week iteration with three one-week iterations within it. For the iteration studied, the developers split themselves into three streams, and each stream took responsibility for a discrete set of stories to work on during the three-week iteration. Each stream self-organised during the three-week iteration.

The study covered 8 working days, and was conducted at the XP team's offices. The data gathered included extensive field notes, photographs, and copies of work artefacts. The observers did not intervene in the team's day-to-day activity, but sought to observe normal activity in its natural setting. This involved attending meetings, shadowing members of the team, going for lunch, and having informal conversations with team members.

2.2 Distributed cognition

Distributed cognition [5] is a theoretical account of the distributed nature of cognitive phenomena across individuals, artefacts and internal (i.e. cognitive) and external representations. In short, it views collaborative work as one cognitive system. The framework has been used to interpret qualitative data such as that collected through observational field studies and interviews in the context of ship navigation [5], aircraft piloting (e.g. [9]) and call centres (e.g. [10]). It has also been adopted by researchers in HCI (e.g. [11, 12]) and CSCW (e.g. [13]) as a useful tool for analysing collaborative work with the aim of identifying breakdowns and answering 'what if' questions, particularly around the use and impact of technology.

The use of distributed cognition to understand software development activity has been limited. The

most-cited work is that by Flor and Hutchins [14], who used distributed cognition to analyse the cognitive system at work in a software maintenance task.

Typically, a distributed cognition analysis results in an event-driven description which emphasises information and its propagation through the cognitive system under study.

Whilst it has the potential to shed light on several aspects of XP teamwork, distributed cognition has not been used very widely. The development, from distributed cognition, of DiCoT, a methodology and representational system to support the distributed cognition analysis of small teams [7], is a promising approach which is explored in this paper.

2.3 DiCoT (Distributed Cognition for Teamwork)

To help us apply distributed cognition to our data, we have adopted the DiCoT methodology. This methodology draws on ideas from Contextual Design [15] and uses representations adapted from Contextual Design, together with a series of principles that are central to distributed cognition. It has been developed and applied in the context of an ambulance control centre, but has not been applied to software teams before.

There are three main themes used in DiCoT:

1. The *physical theme* focuses on the physical environment within which the cognitive system operates. This is important from a distributed cognition perspective because the physical arrangement of the environment, including what an individual can see and hear, affects their cognitive space.
2. The *artefact theme* focuses on the detail of the artefacts that are created and used in carrying out the activity under study. These are important in distributed cognition because they are an integral part of the cognitive system and how it operates.
3. The *information flow theme* focuses on what information flows through the cognitive system, the media which facilitate that flow and how the information is transformed in the process.

Furniss and Blandford [7] identify 22 principles of distributed cognition which can be loosely categorised according to these three themes (see Table 1). Each theme can be investigated using these principles, an associated model, and a tabular representation to capture the detail of activity within a theme. Two aspects of distributed cognition that have not been expanded within the DiCoT method are evolution over time and the role of social structures in coordinating activity, but we include the principles in Table 1 for

completeness, and discuss these principles in our analysis.

Table 1 The principles of distributed cognition underlying DiCoT

Physical Layout
Space and cognition: considers the use of space to support activity, e.g. laying out materials
Perceptual: considers how spatial representations aid computation
Naturalness: considers how closely the properties of the representation reflect those of that which it represents
Subtle bodily supports: considers what if any bodily actions are used to support activity, e.g. pointing
Situation awareness: considers how people are kept informed of what is going on, e.g. through what they can see, what they can hear and what is accessible to them.
Horizon of observation: considers what an individual can see or hear (this influences situation awareness)
Arrangement of equipment: considers how the physical arrangement of the environment affects access to information.
Information flow
Information movement: considers the mechanisms (representations and physical realisation) used to move information around the cognitive system
Information transformation: considers when, how and why information is transformed as it flows through the cognitive system
Information hubs: are a central focus where information flows meet and decisions are made.
Buffers: hold up information until it can be processed without causing disruption to ongoing activity.
Communication bandwidth: considers the richness of a communication channel, e.g. face-to-face communication imparts more information than email
Informal and formal communication: recognises that informal communication can be very important
Behavioural trigger factors: cause activity to happen without an overall plan needing to be in place.
Artefacts
Mediating artefacts: are used to perform the activity
Creating scaffolding: considers how people use their environment to support their tasks, e.g. creating reminders of where they are in a task
Representation-goal parity: considers how artefacts in the environment represent the relationship between the current state and goal state.
Coordination of resources: considers the resources (e.g. plans, goals, history and so on) that are co-ordinated to aid action and cognition.
Evolution over time
Cultural heritage: considers the elements of the

environment that have built up over time.
Expert coupling: considers the information processing cycles in the cognitive system – as an individual becomes expert, these become faster
Social structures
Social structure and goal structure: considers how the social structure within the team relates to the goal structure
Socially distributed properties of cognition: considers how the cognitive system is distributed within the team.

3. The DiCoT analysis of our XP team

The focus for our analysis was the flow of information through, around and within the XP team. In order to analyse this, we focused on the information flow theme, but we also analysed the cards and the Wall, as key mediating artefacts, and the team's physical environment as affecting information readily available to the team. We present these analyses below. Space precludes us from including all the details of each model and the tabular descriptions; instead we present representative extracts. Where possible, we include photographs in order to illustrate how the model is realised in practice.

3.1 The physical theme

The physical theme covers all aspects to working which have a physical layout component, but we have focused here only on the physical layout of the team's environment since this has considerable significance within a distributed cognition analysis (see Table 1). Focusing on the team's physical environment helped us to orient our discussions of later models, and to get an overall view of the team's working space. This physical model is shown in Figure 2 (a).

Seven principles relate to the physical theme. From Figure 2(a), these principles provide several insights relevant to our focus on cards and the Wall (although two of the principles, naturalness and subtle support of bodily parts, did not relate to our model).

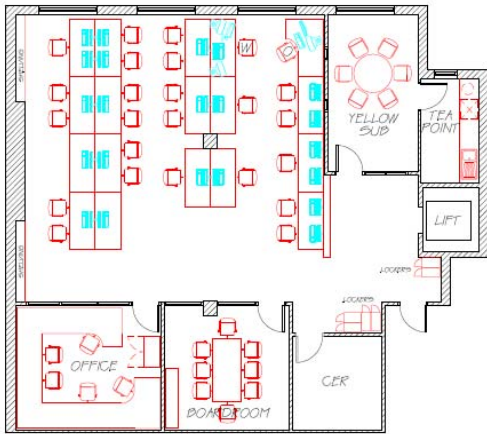


Figure 2(a) The physical model of the team's environment



Figure 2(b) The team's office environment, taken from the bottom right of the physical model in 2(a)

Perceptual: the visibility of the different coloured cards (green, white and pink – see Figure 4(b)) on various parts of the Wall gives developers an overview of what needs to be done from a distance.

Space and cognition: the team make extensive use of their surroundings, in particular the walls of the room. Each stream has its own Wall and so three of the wall spaces are taken up with story and task card displays. A further wall contains other information such as previous iteration summaries (what work was completed in previous iterations), and bug cards (pink index cards describing bugs) and important information such as client contact numbers (see Figure 3).

Situation awareness: the office environment makes everyone very aware of the current situation. Firstly, there are progress indicators from the stream Walls and the bug wall that are visible everywhere within the office except the kitchen. Secondly, the team are in close proximity to each other and so can overhear issues being discussed around them. Third, there are two meeting rooms, both of which are next to the

developer area and if clients come to visit then everyone is aware that they are in the office.

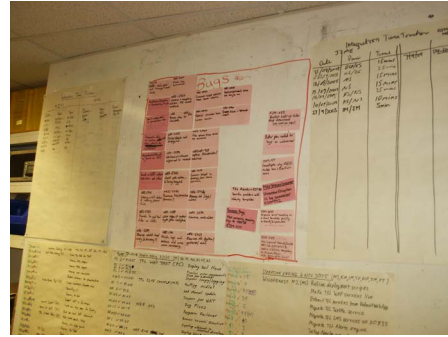


Figure 3 The bug wall (note that this does not have the same structure as the stream Walls)

Horizon of observation: every individual in the developer area can easily see at least two Walls, the bug wall, the meeting rooms and the business development room. Developers are also within easy listening of each other's conversations, questions and advice. However the business development staff are located in a separate room (with the door open) located in the bottom left of the physical model in Figure 2(a). This means that the business development conversations are not so easily overheard.

Arrangement of equipment: apart from the index cards and the Wall, the equipment used includes the developer machines, and the occasional book. Developer machines are distributed around the office with some stations set up for pairing and others set up for single working. This meant that individuals would sit at particular desks in order to achieve a task, e.g. checking their own email, but would move around to attend stand-ups and take part in pairing. The office environment was quite dynamic, with people often moving around.

3.2 The artefact theme

All the cards used by this team – story, task and bug – had the same structure, and were distinguished by different colours: green for story, white for task and pink for bug. The artefact model of this card, as used by our team is shown in Figure 4(a). This model is built up from the cards we saw rather than from any template or company-prescribed format. The artefact model for the Wall is shown in Figure 5(a). Figure 1 shows a photograph of one of the Walls in use by the team.

Each of these artefact models has associated with it a box containing an extract from the more detailed description captured through the tabular format (see Figures 4(b) and 5(b) respectively).

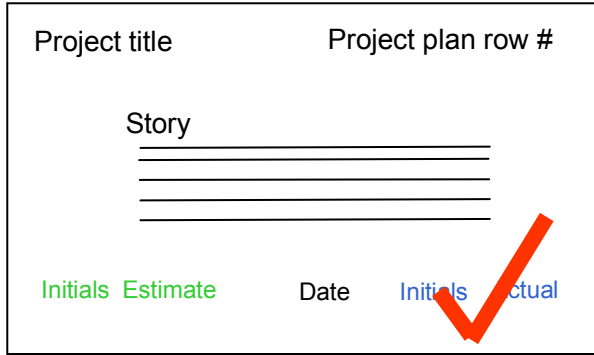


Figure 4 (a) The story/task artefact model

SUMMARY

The story card is a 3inch by 5inch index card. It is a mediating artefact within the system. It contains information for planning, and is the focus of decision-making and code production. Each story card (which is green) may be broken down into several task cards (which are white). Both story and task cards have the same artefact model. Bug cards (pink) are generated whenever a problem is found with the system.

DETAILS

Each card is associated with a project (in the top left hand corner) and this is related directly to the company's overall work plan. Each story corresponds to a line in the company project plan spreadsheet, and the number associated with that line appears on the right hand side of the card (sometimes at the top, and sometimes at the bottom, and sometimes in between). This number is present on story cards and task cards.

Cards are created during the iteration planning meeting, the stream planning meeting, the stream stand-up or during pairing.

The story or task itself is written in the middle of the card. At the bottom left, in green, is written the estimate in days and the initials of the person who suggested that estimate. At the bottom right, in blue are sets of initials showing who has worked on the story/task, together with the date and how long they have spent on it. When the story/task is finished, a large red tick is written in, often over the top of the blue writing, but not always.

Figure 4 (b) An extract from the tabular description of the story/task artefact model

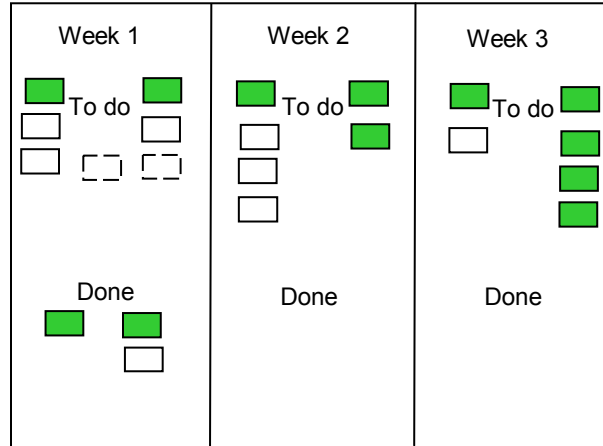


Figure 5 (a) The artefact model for 'The Wall'

SUMMARY

The Wall in this company is a glass screen which is used to display the cards being processed in this three-week iteration. The organisation of the Wall is such that an impression of progress can easily be gleaned. Each stream has their own wall.

DETAILS

The Wall for each stream is structured slightly differently, but the principle is the same in each case. Here we describe the Wall based around the model as shown. The cards are divided into areas representing the three weekly iterations. Within these weekly iteration areas, the cards which have not yet been implemented are placed at the top of the area, and any task cards are placed immediately below the associated story card. Any cards that have been completed are placed at the bottom of the iteration area.

As cards are taken down to work on, a 'ghost' of the card is drawn onto the glass so that the card's place on the Wall is maintained. When the card has been implemented, it is returned to the Wall.

At the end of a weekly iteration, cards that have not been completed are moved across to the next week, unless it is the last week in which case they are noted as being unfinished. This information is captured on a summary sheet (not included here) and fed back into the iteration planning meeting.

Figure 5 (b) An extract from the tabular description of the 'Wall' artefact model

Four principles relate to the artefact theme.

Mediating artefacts: we have already identified the cards and the Wall as key mediating artefacts.

Creating scaffolding: elements of both the card and the Wall provide scaffolding to support the team. The annotations on the cards give teams a clear indication of how far the work has progressed, and who has progressed it; and how much time is estimated to be left can be calculated easily from this information. The Wall, because of its structure, also gives teams a clear idea of progress. Drawing 'ghost' cards on the glass when a card is being worked on by a pair means that even when a card is not present on the Wall a complete picture of progress is still available. *Representation-goal parity:* the desired state for all the stream Walls is for the cards to be in the bottom half of the space, i.e. for all cards to have been 'done'. Of course in this case, the real goal state is for correct code to be developed, and this is represented transiently by the occurrence of 'green bars', i.e. indicators in the Java environment that code has passed all the tests. However, this does not represent a complete or persistent overview; the combination of Wall and card structure can provide such information for the duration of the iteration.

Coordination of resources: the Wall provides key co-ordination. It captures planned work (all the cards to be done), the history of work (what cards have been done), and goals (that all cards should be in the bottom half of the space). The cards, which make up the Wall, capture plans (estimates), history (who worked on the card) and goals (the red tick) as well. The plan for the three-week iteration does show elements of ordering, by placing cards in each of the three one-week iterations. However the plan for a week is not structured, i.e. when the cards are placed on the Wall, no ordering is imposed. The team decides which cards to work on in what order at their daily stand-up meetings. This stand-up takes place in front of the Wall, using it as a focus of activity.

The Wall is constantly updated and acts as a reminder of conversations and design decisions as well as progress. It is also a form of dynamic documentation, where the documentation is not extra work but is integral to the act, i.e. when a developer places the card back on the Wall, they choose where to put it and that choice communicates significant information.

3.3 The information flow theme

We focused on story cards and produced the information flow diagram in Figure 6. For definitions of information hub and buffer, refer to Table 1.

This diagram highlights the key role of the story cards and the Wall. They act as a bridge between the

different activities that the XP team engage with. However this diagram does not capture the detail of the activity, nor the information flows between people. So, the following is a more detailed description of these activities (which in DiCoT are normally captured in tabular form). In order to illustrate how the cards, the Wall and the identified information hubs and buffer operate together, from an information flow perspective, we produced an overall representation, which is shown in Figure 7. In the description that follows we have emboldened the elements that are included in Figure 7.

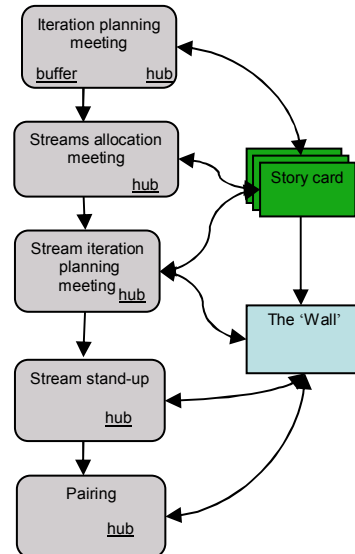


Figure 6 Overall information flow for cards

The **iteration planning meeting** takes place once every three weeks. This involves the stream heads, the project manager and business development staff, and they identify how much time will be devoted to each project over the next three weeks. To do this, they focus on the company's workplan which shows all their clients and the work to be done (as a spreadsheet). Other members of development staff or business development are called into the meeting to provide specialist advice or comments as appropriate. This meeting is both a buffer and an information hub, as some stories are chosen for further work while others are left to another iteration, and decisions are made as to what work will continue.

The **streams allocation meeting** takes place after the iteration planning meeting, and this involves identifying how to group the work identified into streams, and which developers will work in which stream. This is an information hub.

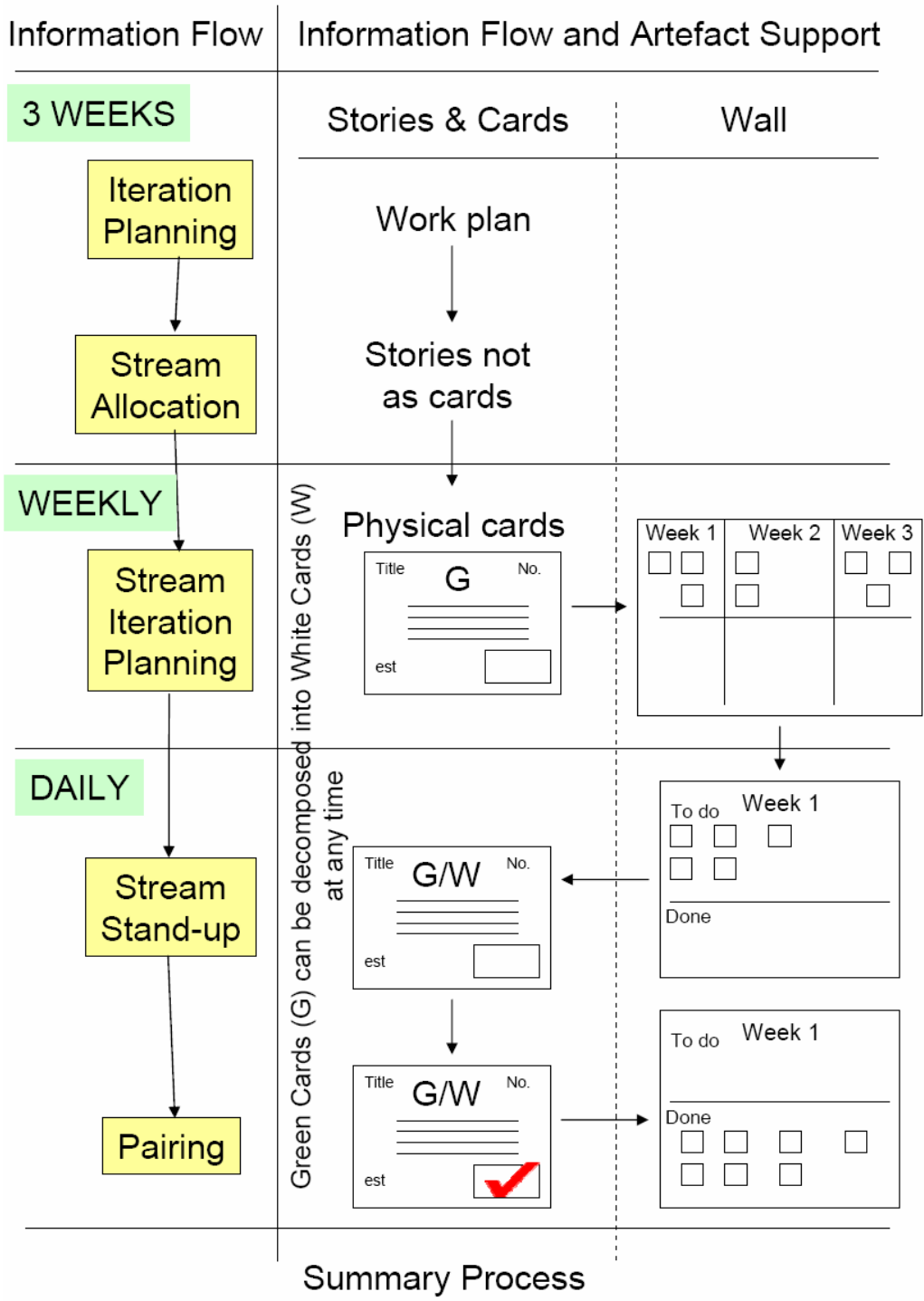


Figure 7 Overall information flow for cards Note that story cards are green (G) and task cards are white (W). Where a card might be a story or a task, this is denoted as G/W

Each stream holds a **stream iteration planning meeting** to work out how they will spread work on the cards across the three week iteration, i.e. which stories and tasks will be tackled in which of the three one-week iterations. New cards are generated during this time, as necessary. The Wall is created during the first week's stream iteration planning meeting. Stream iteration planning meetings for subsequent one-week iterations review what progress has been made in their three-week plan so far, and how to handle any cards remaining. This meeting is an information hub.

The **stream stand-up meeting** takes place every morning and is a key point of knowledge sharing and situation awareness. Each developer reports on what they did yesterday, any problems they encountered and any issues others should be aware of. During these reporting sessions and any subsequent discussions, a lot of information regarding the business of the team is exchanged, but rarely captured, except by updating the cards. The work for the day is planned and pairs are formed to tackle the cards. This meeting is an information hub.

Pairing involves two developers working on a card which they take down from the wall (having drawn a 'ghost' card). When a pair is working they draw on many information sources and the information flows round them are very rich, and immediately applicable, i.e. they don't have to wait for relevant information in order to get on with their work. The information flows around a pair coding are modelled in Figure 8.

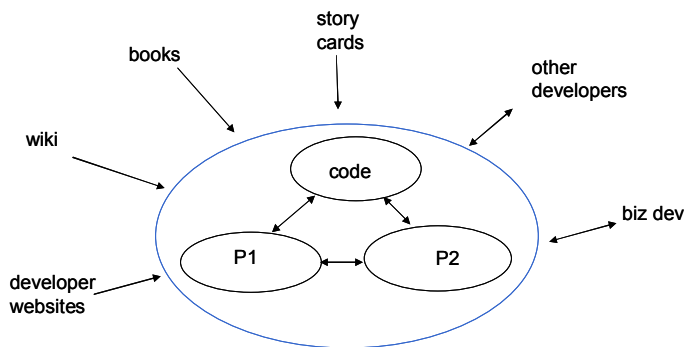


Figure 8 Information flows around a pair

Building on the physical model, we can consider how the cards themselves are moved physically within the environment. This is shown in Figure 9, where the card is taken from a wall to a pair's desk, where it is worked upon, then returned to the wall either in the same place (if it is not completed) or to a new place on the wall, if it has been 'ticked off'.

This open movement of the cards improves situation awareness as it is a resource that everyone shares.

Information movement: the key mechanism used to move information around the cognitive system is face-to-face interactions. XP developers spend most of their time in meetings; we include pairing as a kind of meeting. We have commented above that physical cards move along a simple and open path, but this physical movement is to facilitate pair working, e.g. by making relevant information easier to read, rather than to move information through the system. The Wall radiates information and the act of manipulating the cards in a 'public' way makes activity both visible and shared.

One other mechanism for transmitting information is the company's internal wiki where information about code modules, company procedures, and news items are posted. This is used particularly during pairing, and is one of several information sources available to developers.

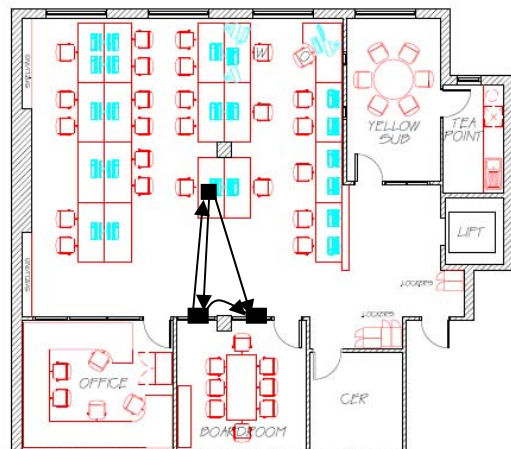


Figure 9 Physical movement of the cards is open

Information transformation: the key information transformation that takes place is turning cards into correctly executable code and this involves a lot of information sources (see Figure 8). Other than this, the main transformations are between people, and between people and cards, and between people and the Wall. The Wall implicitly holds information about progress, and that information is transformed regularly from people through the location of the card on the Wall, i.e. is it 'to do' or is it 'done'? Information is crystallised in the company wiki periodically as appropriate.

Information hubs: Figure 6 indicates five hubs, four of which are where planning decisions are made: the iteration planning meetings, the streams allocation meeting, the stream iteration planning meeting and the stand-up. In each of these, information sources are brought to bear on decisions to be made - mostly about

which cards will be developed when. These information hubs tend to have a central focus of information, e.g. the workplan, the Wall and cards, which people who attend the meeting work on and transform. Other sources of information are the people who attend the meeting.

Pairing is also marked as an information hub, but it is quite a different kind of activity from the others. Pairing brings together several information sources to produce code and along the way, planning decisions and design decisions are made.

Buffers: the workplan is a kind of buffer, and the Wall also holds up information until it can be processed. The workplan, however, is not widely visible to the team, unlike the Wall.

Communication bandwidth: bandwidth of communication is generally high as communication relies heavily on face-to-face interactions. The physical size of the cards means that there is a limit to the amount of information they hold and so developers are encouraged to talk with business development staff and each other in order to know what needs to be implemented.

Informal and formal communication: this cognitive system abounds with informal communication channels. Particularly within the pairing activity, developers are constantly communicating in an ad hoc fashion – overhearing others’ conversations, calling on other developers for their expertise, catching a member of the business development team as they walk through the development area to the kitchen, and so on. Meetings such as the stand-up are held regularly (every day) and have a prescribed format, and in this sense are ‘formal’, but they do not have a formal agenda and do not follow rigorous rules.

Behavioural trigger factors: on the one hand, there are behavioural trigger factors everywhere – a card on the Wall causes a developer to produce code. On the other hand, the team follows a rhythm of activity that does not need triggers. This rhythm has been discussed elsewhere [1].

3.4 Time and social principles

Table 1 listed four further principles which have not yet, in the DiCoT method, been elaborated. However these principles are relevant in our context and so we discuss them in this section.

Evolution over time

Cultural heritage: there are different levels from which this principle may be viewed. The wider XP community has built up a considerable set of techniques and traditions, many of which are reflected in the practice of the team under study. At a more local level, the team’s use of the cards and the Wall have

evolved over the life of the company (around 6 years at the time of the study). When the company first started, they used index cards to capture stories and tasks. However they then moved to offices where it was not possible to pin things onto the walls and so they started to use a summary sheet instead. This sheet captured similar information to that on the index cards, but it listed the stories and tasks, together with estimates and actual times, rather than having them on separate cards. These summary sheets are still in use today and are the focus of a company-wide meeting which takes place at the start of each three-week iteration (not featured in Figure 2).

Expert coupling: the level of expertise within the company as a whole is quite high, and new recruits are quickly brought up to speed through pairing and an inclusive culture.

Social structures

Social structure and goal structure: both the social and the goal structure were very flat within the company, which contributed to the culture of shared responsibility, e.g. stream heads were understood more as representatives of the stream rather than the bosses, and it was not unusual for recent recruits to chair company meetings.

Socially distributed properties of cognition: the cognitive system for this team is clearly distributed across people. One example of distributed problem-solving is pairing. Here the activity of developing code is distributed not only between two individuals, but also between the various information sources identified in Figure 8. In our observations it was quite common for a pair to refer to a fellow developer when appropriate, or even for the pair to turn into a three-some and then into a pair again, but with different developers. This distribution extended to all members of the company including non-developers. Anyone in the team may be called upon to help in resolving a problem. The socially distributed nature of problem solving in XP is discussed further in [16].

4. Discussion

4.1 Some observations

From the analysis reported above, the following observations emerge:

1. There are few mediating artefacts in the system and hence there is little transformation between different media. In addition, the role of the mediating artefacts is largely restricted to process issues. For example the cards and the Wall aid in handling plans and goals but lack detailed information. This means that no artefact

substitutes for more meaningful high-bandwidth discussion between people. In fact, the scarcity of information encourages this behaviour.

2. The information flows are simple and open, thus promoting situational awareness amongst the team (e.g. see Figure 2)
3. The XP team works in an information-rich environment. This is particularly evident through our modelling when considering the information flows around the pair. Information is both easily accessible and immediately relevant and applicable.

4.2 Interrogating the themes

Here we pose two example ‘what if’ questions regarding information flow within the team, and use the analysis above to provide some answers.

If the Wall and the cards were changed in character, what effect would this have?

For example, if the Wall was kept in a box on the stream head’s desk, with compartments for each week, and sections to show the ‘to do’ cards and the ‘done’ cards, how would this affect the information flows within, through and around the team? The key difference here would be that the Wall is no longer easily accessible or visible. The Wall would still create scaffolding to support the work, reflect the goal state of the work and co-ordinate resources. Stand-ups would need to be conducted around the box, or with the cards spread out (but then there’s extra effort in taking them out and putting them away). The Wall’s ability to support co-ordination would therefore be diminished.

Having the Wall as a pile of cards would also affect situation awareness as these progress indicators would no longer be visible. It would no longer be an information radiator and hence there would be less information moving around the cognitive system.

The transformation of information between people and the Wall would still be necessary and would be supported by the compartments within the card box.

So it seems if the Wall were to take this form, then the key difference would be in situation awareness and to an extent in co-ordination of resources. We speculate that this would have a detrimental effect on the team’s cohesion.

What if the Wall was in electronic form? For example, the cards could be stored in a database, and then displayed automatically in the Wall format. It would not be possible to physically handle the cards themselves, but such handling could be achieved electronically. If the system supported collaborative working then each member of the team could move the cards around on the Wall, just as they do now. The

only difference would be that they are moving electronic representations rather than physical cards.

Drawing ghost cards would need to take another form, but could be implemented using a variety of means including ‘locking’ the card.

With an electronic version of the Wall (and hence of the cards), information flow may be less visible. If the team had a large plasma screen displaying the Wall, then this would make the Wall more visible. However this approach would obviate the need to physically manipulate the cards in a public fashion – no-one would be seen to approach the Wall and modify its display. This would affect situation awareness, for both the stream Walls and the bug wall. At the moment, when a new bug card is put on the wall, other developers notice this and groan. Hence the manipulation of the artefact (and its implications) is shared as well as the artefact itself. We speculate that removing this sharing would affect the cognitive system too.

In another team we visited, the cards, tests and progress indicators were stored in a database [16]. One of the tendencies here was to add more information to the system and to link information together. While it’s difficult to identify the cause, we observed a significant breakdown which could be attributed to the electronic nature of their Wall. Shortly after our visit, this team reverted to physical cards.

If you replace pairing with single programmer activity, what happens to the information flows?

If you look at Figure 8, there are three members of the conversation that makes up pairing: the two programmers and the code. Each of these has a significant part to play in creating the code (for more explanation of pairing being a form of conversation, see [17]). Between them there are three equal flows of information. If you remove one of the programmers then the information flows drop to just one flow. This means that they are cut by two thirds, so although the environment is still rich in information, the discussion around developing the code reduced significantly.

The fact that pairs change regularly also has significance for the propagation of knowledge within the company, which in turn makes the company more robust and stable over time. Removing the pair would lessen this effect and may result in the company being less stable.

5. Future work

We see future work within two main areas: continuing the existing analysis of this XP team, and extending the analysis to consider other teams.

The work presented here is only one part of a full distributed cognition analysis of the XP team. We have focused only on the information flows through, within and around the XP team, specifically looking at the cards and the Wall. However there are other aspects that could be analysed. For example, there is a lot of knowledge and information being exchanged between people. The analysis presented here doesn't give any detail about those interactions because its level of granularity is too high. To get a clearer view on what happens within these interactions, a more detailed analysis of conversations taking place between people, both in meetings and in more informal interactions, needs to be conducted.

To understand which aspects of our analysis are peculiar to this one team, and which may be generalised, we need to analyse other XP teams using this approach.

6. Summary

This paper has presented a limited example of the kind of analysis possible using the distributed cognition framework. In our analysis we have found that the framework insists that the details of interaction be attended to. This in turn brings with it different kinds of insight. In particular, the possibility of interrogating the analysis results using pertinent questions holds great promise.

In this paper we have considered the implications of the Wall and cards taking a different form, and have found that either changing the Wall to be a box of cards or using an electronic version of the artefacts would have consequences for the cognitive system underpinning the XP team's work. We have also illustrated the impact of single programmer working rather than pairing, on information exchange.

We plan to pursue further analyses using this framework, both with this team and with others.

7. Acknowledgements

We would like to thank our collaborator company, Kizoom Ltd, for their support and co-operation.

References

[1] Sharp, H. and Robinson, H. (2004) 'An ethnographic study of XP practices', *Empirical Software Engineering*, 9(4), 353-375.

- [2] Highsmith J. (2002) *Agile Software Development Ecosystems*. San Francisco: Addison-Wesley.
- [3] Beck, K and Andres, (2005) *Extreme Programming Explained: Embrace Change* (2nd edition), Addison-Wesley.
- [4] Cockburn, A. (2002) *Agile Software Development*, Addison-Wesley
- [5] Hutchins, E. (1995) *Cognition in the Wild*, Cambridge MA: MIT Press.
- [6] Blandford, A. & Furniss, D. (2005) DiCoT: a methodology for applying Distributed Cognition to the design of team working systems. To appear in Proc. DSVIS 2005. Springer: LNCS
- [7] Furniss, D. and Blandford, A. (2005) Understanding Emergency Medical Dispatch in terms of Distributed Cognition: a case study, to appear in *Ergonomics Journal* Special Issue on Command and Control.
- [8] Furniss, D. (2004). Codifying Distributed Cognition: A Case Study of Emergency Medical Dispatch. MSc Thesis. UCLIC
- [9] Hutchins, E. and Klausen, T. (1991) Distributed cognition in the cockpit. In Engeström, Y. and Middleton, D. *Cognition and communication at work*. Cambridge University Press.
- [10] Halverson, C. A., (2002) 'Activity theory and distributed cognition: Or what does CSCW need to DO with theories?' *Computer Supported Cooperative Work*, 11:243-267.
- [11] Hollan, J. Hutchins, E., Kirsch, D. (2000) 'Distributed Cognition: Toward a new foundation for human-computer interaction research', *ACM Transactions on Computer-Human Interaction*, 7(2), 174-196.
- [12] Wright, P.C., Fields, R.E. and Harrison, M.D. (2000) 'Analyzing Human-Computer Interaction as Distributed Cognition: the resources model', *Human-Computer Interaction*, 15, 1-41
- [13] Hoadley, C.M. and Kilner, P.G. (2005) 'Using Technology to Transform Communities of Practice into Knowledge-Building Communities', *SIGGROUP Bulletin*, 25, 1, 31-40.
- [14] Flor, N.V. and Hutchins, E.L. (1992) 'Analyzing distributed cognition in software teams: a case study of team programming during perfective maintenance', *Proceedings of Empirical Studies of Programmers*, 1992
- [15] Beyer, H. & Holtzblatt, K. (1998) *Contextual Design: Defining Customer-Centered Systems*, Morgan Kaufman, San Francisco
- [16] Sharp and Robinson (2006) A distributed cognition account of mature XP teams, submitted to XP2006
- [17] Robinson, H. & Sharp, H. (2005) 'The social side of technical practices', in *Proceedings of XP2005*, LNCS 3556, 100-108.