Abstract
Agile software development relies on effective communication between project stakeholders. The interaction between the customer and developers is particularly key to ensure that the system under development is usable and used. Co-located teams rely on two simple paper-based artefacts used in sophisticated ways to support interaction, but in a distributed setting these paper-based artefacts need to be replaced. This paper describes one distributed agile team’s simple solution to this problem. Although their approach works well, it may be reliant on several characteristics of the team and the system they are working on.

Keywords
User story, wiki

ACM Classification Keywords
D.2.1. Software Engineering requirements/specification; K.6.1 Project and people management

Introduction
Agile development is a new approach to software engineering that explicitly champions an active role for the customer [1, 2, 5]. Indeed, eXtreme Programming (XP), one of the most popular agile development
methods, strives to include a real user(s) in the team. The customer role has proved to be problematic, even in a co-located situation [7], but with the rise in popularity of distributed development, further challenges need to be faced.

One of the central values of an XP team is ‘simplicity’ and so any solution needs to be the simplest possible. In the case study described below, we explore the simple support solution the team has chosen for their situation, and consider why it works so well. In the rest of this introduction we explore the role of the customer in XP, describe ‘the story’ which is the key unit of interaction between stakeholders, and compare participatory design and XP.

Agile development
Underlying the agile approach is a set of principles and a manifesto which emphasises the importance of "Individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan" (http://www.agilemanifesto.org/). Software is evolved through short time-boxed cycles of planning, implementation and testing, with only minimal documentation, where each iteration may be as short as one week. As a consequence, effective communication between developers and other project stakeholders is key. In particular, interactions between the agile ‘customer’ and developers in order to communicate, discuss and evaluate the emerging system have been the subject of much debate and uncertainty.

The Customer Role in XP
In XP, the customer role is an onerous one. Customers are expected to become an integral part of the development team and to perform a number of tasks including [1]:

- Generating requirements (called stories in XP) and acceptance tests;
- Answering developers’ queries and discussing the detail of stories with developers to ensure understanding;
- Steering the product by setting priorities. This involves making decisions on scarce resources, and can be very difficult to achieve;
- Providing feedback on iterations and facilitating emergent requirements.

In an ideal XP world, the people filling the customer role would be co-located with the developers; would 'speak with one voice'; and would be potential users of the system.

Of course, this ideal is rarely realized [8]: client organisations may be unwilling or unable to spare people to become part of the development team; different customers may have conflicting requirements; potential users of the system may not have the authority to make decisions concerning the identification and prioritisation of system features whereas decision makers may not understand the needs of users, and so on.

Stories in XP
Stories are the key unit of communication between stakeholders in an XP project. During development phases, stories are small units of functionality which can usually be developed after a day or maybe two
days’ effort. Earlier in the product’s lifecycle, ‘epic stories’ and larger chunks of functionality will have been identified and refined. Ron Jeffries explains that each story has three parts: Card, Conversation and Confirmation.

Stories are usually written on 3x5” index cards. The physical size of an index card places a constraint over the amount of information that an index card can hold and so keeps written requirements a brief as possible. Index cards are also small, independent entities that can be moved around and annotated. This helps to facilitate meeting dynamics where focus is clearly on one story or another rather than a page of intertwined requirements. A widely used template for XP stories is:

As a <role>, I want <behaviour> so that <benefit>

Stories are not stand-alone written documents; they are understood in the context of a two-way conversation between business and development teams. This conversation is critical, and is where the stakeholder interacts with the developers in shaping the final product.

An example of user action needs to be discussed as part of the story so that developers can drill down to enough detail to implement and test the story. A user acceptance test is agreed that will be used to confirm that the software development satisfies the user need it represents.

Agile development and participatory design

Participatory Design (PD) is a technique that promotes the direct participation of stakeholders (especially end-users) in system analysis and design. Agile approaches support this goal. The key mechanisms used to support interaction between stakeholders and developers are two paper-based artefacts called the story card and the Wall. To date, there has been little work in understanding the significance of their paper-based nature, and distributed XP teams are suggesting solutions of their own. The case study below examines one team’s experience and their approach.

The Case Study

This case study is based on a short empirical investigation of one agile team within a larger organization. The investigators spent three days observing and interviewing the developers, attended one of the planning meetings with the customer (held by telephone), and reported the observations back to the team for discussion and confirmation.

The Team

The organization within which our study team sat is in the business of providing voice and data services around the world, it is building a “new wave” business based upon networked IT services, broadband and mobility and is divided into several businesses, one of which focuses on software development. The organization employs approx 100,000 staff, about 7500 of whom are software developers. Their headquarters is located in the UK, although a large portion of development work is carried out off-shore.

The team consisted of five developers, one main customer (who represented a large department), and several testers. The customer was located within the UK but several hundred miles away. The testers were located off-shore. All developers were co-located (see Figure 1).
The system they were working on was being used internally by the network management department within the same organization. It was a live system and the team were working on various upgrades and enhancements.

Stakeholder interaction – the wiki
The central focus for the co-located developers is the physical wall and story cards (as observed in other teams). See Figure 2.

The central focus for the distributed team, i.e. the developers plus the customer and the testers, is a wiki, which replicates the relevant characteristics of the physical artefacts.

Interaction between the customer and the developers routinely takes place once a week, at the start of a development iteration. This meeting is held by telephone, and is supported by the wiki. In addition, the team members call the customer if particular queries arise during the iteration week.

The wiki used to support the team includes background information about the system and the team, and the history of the development. The section of the system that supports day-to-day interaction is a set of pages which records the stories to be developed during the current iteration, and their status.

The template for these online user stories is given in Figure 3. Note the different colours used to denote the different status. For the physical story cards, different coloured stickers were used to denote the different statuses, but interestingly the colours used on the wiki were not the same as those used on the physical cards.

The wiki is accessible to all team members at any time, and any of them can update the status of a story. Testers work on the stories that are ready to test overnight, and test results are available the next morning.
Lessons Learned
A key characteristic of agile working is simplicity. The situation described here shows how a simple wiki can support the interaction this agile team required. Other studies we have conducted confirm the importance of the story cards and Wall for communication between co-located team members \([8, 9]\). The mixed-mode solution adopted by this team works well and has some interesting characteristics:

1. Information is replicated across the physical and the electronic. When asked, team members were not concerned about this replication of work, nor the possible errors.

2. Members of the development team have worked with the system under development for many years. They claim that they have a good understanding of what is likely to be acceptable to the customer and what is not.

3. Interface design issues were not referred to the customer before being implemented. The customer would see the changes when the system was demonstrated, just before release.

4. It relies on disciplined use by all team members. There are few checks and balances here to confirm what has happened.

Conclusions
Agile development’s short iterations and evolutionary development style relies on effective communication between the stakeholders, including users and developers. In co-located teams, this communication is supported with simple physical artefacts, but in a
distributed situation, other means of support need to be found.

In this paper we have discussed one team who used a simple mixed-mode form of support consisting of physical artefacts for co-located team members, and a wiki for distributed team members backed up by telephone conversations. For this team, the system worked well, but how acceptable would this be in other circumstances, specifically with a more complex system or where the developers have less domain knowledge?

As simplicity is a central tenet of agile software development, any alternatives would need to be simple with (at most) little overhead. Further work in this area will include investigating the communication challenges presented by agile software development, and identifying techniques and methods used by other disciplines and other teams in supporting dialogue between stakeholders.

At this workshop, I would hope to share agile’s experiences of supporting distributed collaboration and learn from others’ experiences in similar situations.

Acknowledgements
I would like to thank the members of the XP team described in section 2 for their support and patience.

References