

Customer collaboration: challenges and successes in practice

An Agile Development 2003 technical exchange

Wednesday, June 25, 8.30 – 10.00, Salt Lake City, UT

Background

This TE session was run in a goldfish bowl format. Approximately 25 people attended for most of the time (there was some moving about). We explored, shared and reflected on our practical experiences of making customer collaboration work in all its various forms. We took a snapshot of the state of practice in customer collaboration, covering the various approaches and techniques that have been used.

The discussion

The discussion covered a wide range of issues which we group under the themes below. The themes are interrelated (e.g. aspects of trust might be supported by a workshop, or the use of models), and were re-visited throughout the discussion as the themes emerged.

Trust

Throughout the discussion, participants emphasised the importance of trust (between developers and customers) as an absolute pre-requisite for any form of customer collaboration. This is about building a language of exchange and it is something you have to keep working on. When you have trust, many other problems go away. Very deep trust is the key to success but you have to build it up. Delivering the 'right' software not just 'working' software is important here. You can only say that you've delivered value when the customer is using the product.

Despite the strong endorsement of trust by everyone, one attendee commented that even when you don't have trust, if everyone is pulling together, with a shared vision, then the project can be successful.

The Bridge

Many participants found the notion of customer collaboration as a bridge between two worlds as a useful way of viewing things. That is, customer collaboration is a role/person who acts as a bridge between the world of the user/customer and the world of the developer. Several participants suggested that software testers were often a central plank in such a bridge role.

A 'bridge' has a variety of skills. A successful bridge is able to build the relationship between the business and the developers. They need to know how the different groups work, and what is important to them. The bridge understands both worlds, and knows the questions to ask of the business which will be relevant to them. This will develop trust between the business and the developers. It was suggested by one attendee that building a trusting relationship may be more important than communication (no-one disagreed). One aspect of this is that ambiguity should be kept away from developers because developers don't want to live with ambiguity.

Testers may make good bridges because they have the right personality traits. The following were suggested: testers are happy to be dilettantes, i.e. shallow; testers are fond of 'unpopular' people such as the customer support people (but they learn a lot from them); testers have a tolerance for ambiguity and lack of agreement.

Modelling

Models and modelling were suggested as a form of bridge, in the sense that the activity of modelling enables a series of questions to be asked that promote shared understanding and meaning – and, ultimately, trust. It was suggested that a model lives between the customer and the developer.

Modelling is a way to communicate and we should use different models - find something that 'sings' to the customer. Modelling skills help you identify ambiguities, and when you break a model you learn. Models are a good way to represent all the information available. XP suggests that you shouldn't have documentation and you shouldn't have models, but what's wrong with having a range of information available to you, including a model? Developers don't have to see the model, it may be kept 'private'. Most collaborators don't understand UML - we need collaborative modelling skills (models don't have to be built in UML).

The correct model is a tool for the bridge role.

Customer Roles

There was wide consensus that ‘customer’ was a catch-all phrase that needed to be pulled apart so that various sub-roles were made explicit. Four such sub-roles were put forward:

usage expert - someone who knows how to make short cuts that work;

domain expert

product owner - avoids the problem of the 'disappearing' customer, or of multiple customers

executive sponsor - someone who has the power to cancel the project

It was suggested that customer collaboration in the XP sense tends to emphasise the first two roles (usage expert and domain expert). However, the importance of the product owner sub-role was also underscored – someone who ‘understands business but can also talk technical’. An individual in this sub-role was seen as a customer surrogate, useful particularly in situations where there were multiple customers or no customer. It was suggested that an alternative for ‘product owner’ was ‘product champion’, suggesting that the sub-roles of product owner and executive sponsor were not always distinct.

Access to customers

Customer collaboration requires access to the right people, but having got access you should ask the right questions; time is precious. Getting customers to write stories is hard, but one attendee commented that having got the stories, getting access to the right people to fill in the gaps in the story was harder. On the other hand, if you have too much time with the customer then you run out of questions to ask.

If customers are passionate about the product, then you will get access. Understanding and emphasising the business value of a product and linking it to the business strategy is important and will help to make them passionate. One attendee believed that getting the business value right was of paramount importance.

Workshops

Several participants reported on the importance of planning participation with customers, stressing that collaboration is not something that will ‘just happen’. Shared, face-to-face space via workshops was seen as highly effective with the notion of ‘requirements by collaboration’. You need different people to take part in this planned participation depending on where in the development cycle you are, and this will also affect what you get out of a workshop.

Sometimes customers don't collaborate between themselves. Workshops can help surface this. Coaching of the business side can also help.

Supporting collaboration

The need to have a business coach as well as a developer coach was seen as important by several participants – balancing a view that XP tends to be developer-centric. So there is a need to support people in the bridge role with, for example, appropriate modelling approaches. Overall, participants underscored the importance of creating a shared community with a shared vocabulary as a key aspect of customer collaboration.

Other areas

Other comments/questions were:

- It was felt that some areas – such as high innovation projects/‘greenfield’ developments – placed particular demands on customer collaboration which still needed exploration.
- Non-agile development teams have artificial deadlines that come from fear of being unable to deliver, but agile projects should not suffer from this because the artificial deadlines go away.
- You need to be able to manage the expectation of the customer.
- Collaboration = negotiation, and if so does it need structure? (this question was left unanswered).
- XP hasn't really looked at the customer role because it's developer-centric.

Useful references:

Linda Rising (2000) 'Customer Interaction Patterns', in Neil B. Harrison, Brian Foote, Hans Rohnert, editors. *Pattern Languages of Program Design 4*. Addison Wesley.

Ellen Gottesdiener (2002) *Requirements by Collaboration: Workshops for Defining Needs*
Addison-Wesley

For more information, please contact Helen Sharp (h.c.sharp@open.ac.uk) or Hugh Robinson (h.m.robinson@open.ac.uk).