Technical Report No: 2004/10

Customer collaboration: successes and challenges in practice. Report of an activity session held at XP 2003

Helen Sharp
Hugh Robinson
Judith Segal

17th March 2004

Department of Computing
Faculty of Mathematics and Computing
The Open University
Walton Hall,
Milton Keynes
MK7 6AA
United Kingdom

http://computing.open.ac.uk

Customer collaboration: successes and challenges in practice
Report of an activity session held at XP 2003

Helen Sharp[§]*, Hugh Robinson[§] and Judith Segal[§]
[§]Computing Department, The Open University, Walton Hall, Milton Keynes MK7 6AA, UK
*& Visiting Fellow, Centre for HCI Design, City University, Northampton Sq, London, UK

Abstract

This paper reports on an activity session held at XP2003 in Genoa, Italy in May 2003. The intention of the session was to collect experience of customer collaboration in XP projects with the intention of identifying common models, their critical success factors and their pitfalls. As reported in the books, XP expects an on-site customer, i.e. someone co-located with the development team who is able to make decisions about required functionality and funding while at the same time representing the real users of the product. Anecdotal evidence has suggested that this model is not widely used, and the results of this session confirm that there are many different successful models used in practice.

1. Background

The Agile manifesto emphasises the importance of valuing "*Customer collaboration over contract negotiation*". An overview of requirements is expected at the beginning of an agile project, but the principle of collaborative partnerships between developers and customers underpins the process of negotiation required to expand the requirements and manage constant change. In XP, this principle is instantiated through the practice of having an 'on-site customer', someone who is available to answer questions, resolve disputes and set small-scale priorities on an on-going basis (Beck, 2000). This customer should be someone who will use the system, i.e. an end user. Others have discussed the practicality of involving end users in software development and while there is general agreement that it is important to know the users, understand their priorities, and actively involve them in identifying requirements (e.g. Gould & Lewis, 1985;Kotonya & Sommerville, 1998), there is dispute about how much user involvement is beneficial (Heinbokel et al., 1996; Keil & Carmel, 1995), and exactly how best to manage user input.

The traditional view of requirements engineering involves a substantial period of investigation and negotiation to uncover the client's requirements (e.g. Robertson & Robertson, 1999) and it has been claimed that the success of a delivered software system can depend crucially on requirements engineering (e.g. Taylor 2000). The role of the customer in XP is different when compared to traditional software development methodologies but is just as crucial to the success of the project. Although issues concerned with agile requirements engineering have been raised in the past (e.g. TCRE02, 2002), little has been done to uncover what happens in practice. For developers using XP, the need to collaborate more closely with the customer presents a variety of issues including: what changes will need to be made to existing customer communication routes? what makes a collaboration successful? who should be involved in the collaboration? where should the customer be located? if it's infeasible to be on-site, what impact will this have on collaboration?

This paper reports on an activity session held at XP2003 in Genoa, Italy, to investigate these and other questions about the nature of customer collaboration in practice, with a view to explicating what works and what doesn’t.

2. Session structure

Nineteen participants attended the session. Of these, 4 said they were new to XP, 6 had less than one year's experience and 9 had more than a year's experience. Following an introduction to the area, the session was structured around two group tasks. In the first task, participants were asked to write down on index cards their experience of customer collaboration in terms of:

- Who represents the customer?
- Where are they located?
- When/How often do they engage with the development team?
- What does the customer do (is responsible for)?

For example, 'classic' XP as described in the books says that the customer is responsible for:

- Writing stories and acceptance tests
- Resolving disputes on priorities
- Stopping irrelevant work
- Making decisions on scarce resources
- Discussing the detail of stories with developers to ensure understanding
- Providing feedback and facilitating emergent requirements
- Steering the product

Having spent about 30 minutes on this task, the cards were laid out on the floor and participants were asked to group the models to identify commonalities.

For the second group task, participants were asked to choose one of the common models and form different groups to discuss the critical success factors, and its pitfalls and avoidance strategies. Participants were asked to consider, for example,

- What makes the collaboration successful?
- How are all the stakeholders represented?
- Are there alternatives to 'on site' customer?
- Is innovation catered for?

These outputs were written on flip chart sheets for presentation during the wrap-up session where each group presented their results to the whole attendee group.

3. Results of the first task and the clustering activity

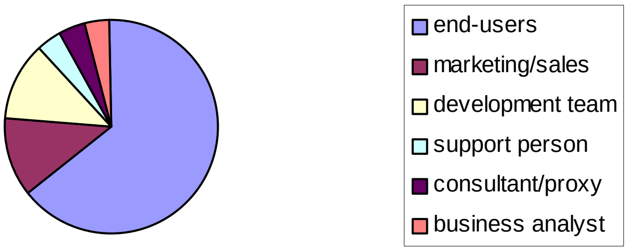
There were 30 projects described on index cards. The following sections present the raw data collected from these cards.

3.1 Who acted in the role of customer?

Here, it was clear that some products were 'shrink-wrapped' (as opposed to bespoke) and some were for the developer's own company (so the roles of developer and customer shaded into each other).

However, of the 30 projects: 2 cards were too ambiguous to include, 16 had end-users as customers, 3 had marketing/sales people, 3 had members of the development team, one had a person who will support the product being built, one had a consultant/proxy, one had a proxy, and one had a business analyst (where the product being developed was internal to the development company) (see Figure 1)

Figure 1: Who acted in role of customer?



For those 16 projects where users were the customers, there is insufficient information to say exactly who these customers were, but in 3 cases, this involved a group of end-users who changed as the focus of development changed. There was 1 instance in which the product would be used by two different groups of users and only one group was represented. There were 2 cases in which the customer was the head (project leader) of the end-users. There were 2 cases in which the customers were potential end-users of the system, one case was described as 2 of the 3 potential customers speaking as one voice, six just said end-user/end-user representative/customer, and one project had a "very responsible customer".

A team including product conceptualisation person ('the one voice'); usability expert; user support; web designer.

3.2 Where are the customers located relative to the development team?

Of the projects where end-users took the customer role, only two were located in the same room, three were in the same building (one specified different floor), two were at the customer's site (1 in same room; 1 in different room), one was 'nearby – 5 minutes walk', one was in the same city, one was in a different country and five were 'not on-site'. See Figure 2.

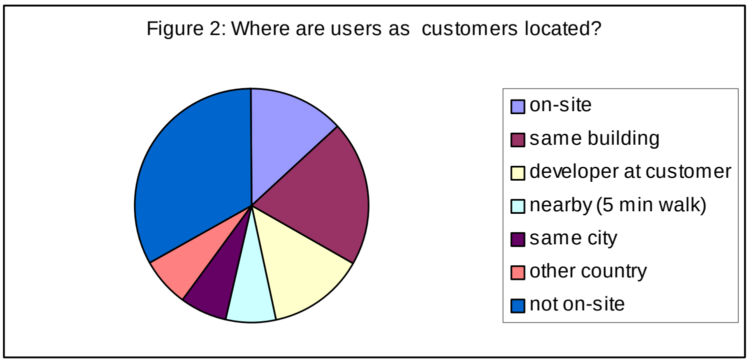


Figure 2: Where are users as customers located?

Of the five 'not on-site', two of them use email and phone to communicate, for one project, the customers call them, for one project communication is via phone, email and weekly meetings, and one project has monthly meetings.

Where 'others' took the customer role (14 projects), seven were located with the development team (see Figure 3), so significantly more of the 'proxies' were on-site than the real users.

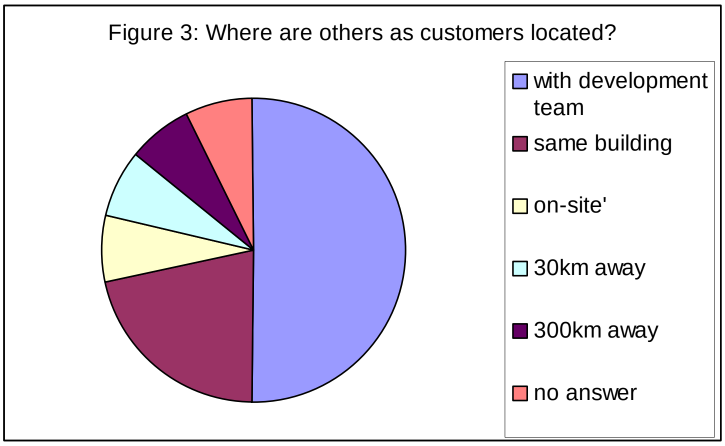


Figure 3: Where are others as customers located?

3.3 When/How often do customers meet with the developers

The picture here is similar whether or not users are the customers. As you might expect, given the co-location of customers with developers where 'others' took the role of customer, there are slightly more frequent meetings than when users take the customer role. In addition, in none of our projects where 'others' were customers did the developers meet with them only once a day (see Figures 4 and 5).

To give some more detail to these figures, where end-users took the customer role, one project had 'continuous' user meetings, and in one case developers could contact the users by phone anytime; one project met several times a day; two several times a week, five one day a week; two every three weeks; and one once a month. Two projects had user-developer meetings several times a day or several times a week depending on the stage of development, while one further project had meetings whenever it was appropriate, again several times a week.

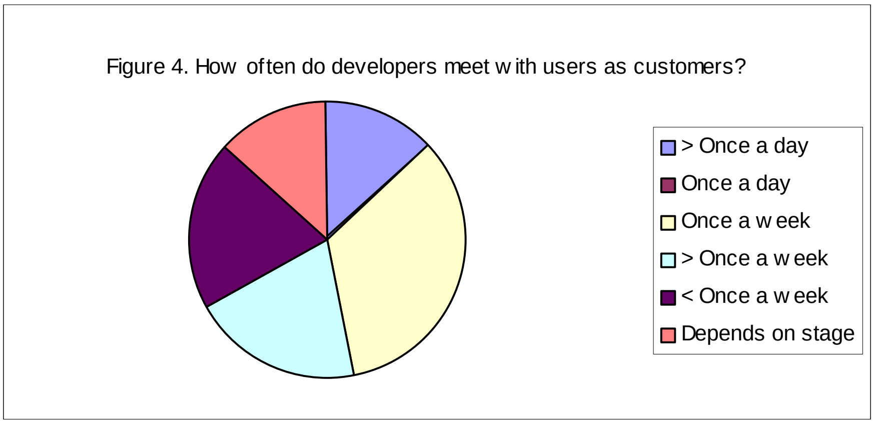


Figure 4. How often do developers meet with users as customers?

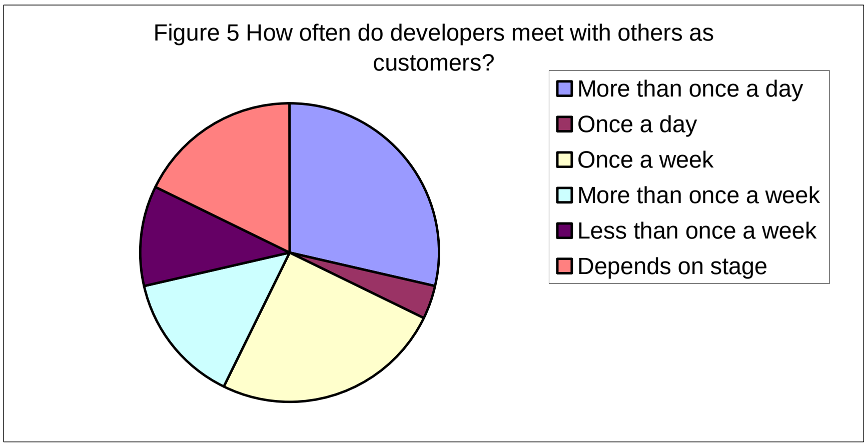


Figure 5 How often do developers meet with others as customers?

Where 'others' took the customer role, in three projects the meetings depended on the stage of development, including one project that met '2 days/week, every day in the last week before release'. Six projects (nearly half of this group) met several times a day, most often through the daily stand-ups backed up with availability throughout the day:

- 'Standup; planning game; during the day to resolve bugs; product clarification; lunch; pairing'
- 'Standup – available for clarification during the day'
- 'Standup; planning game (weekly for several hours); lunch as clarifications came up, problems etc.; sales meetings, in the bar'
- 'Standup – available for clarification during the day – planning game
- 'Did release planning on their own, were involved during iteration planning and accessible through the whole iteration'

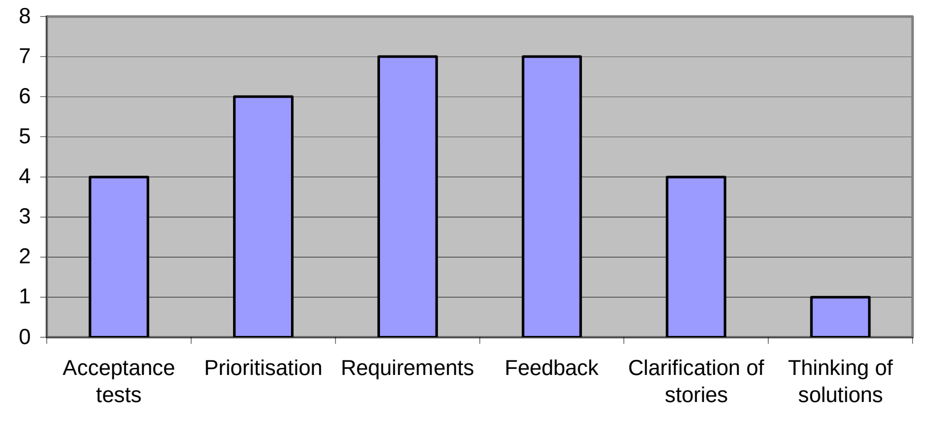
One project held meetings once a day, specifically at the standup meetings; on one project, developers met customers 'everytime the customer has an issue'; one project had user-developer interaction 60% of time; and on two projects developers and customers met once a week but customers were also available by phone when needed.

3.4 What is the customer responsible for?

Of the seven responsibilities listed in the introduction from 'classic' XP books, three of these areas received little or no attention from our projects: stopping irrelevant work, making decisions on scarce resources and steering the product. The first two could be argued to be implicit in 'setting priorities', but the last one could not. Only two of these projects mentioned 'vision' or 'solutions' at all.

Where end-users took the customer role (see Figure 6), four were responsible for acceptance tests and clarification of stories; one was responsible for solutions; six for prioritisation including 'workflow', 'making decisions' and 'steering' (the development, not the product); seven for requirements including 'defining user stories'; and seven for feedback including 'emergent requirements'. These numbers are based on 16 project cards, and some included multiple responses.

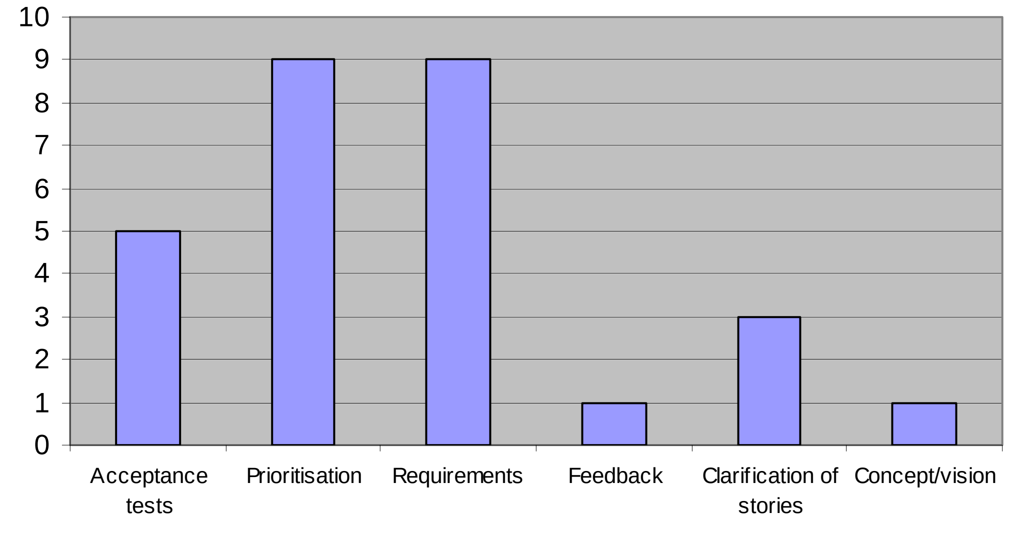
Figure 6 What is user as customer responsible for?



Where others took the customer role (Figure 7) five were responsible for acceptance tests; three for clarification of stories; one was responsible for the 'concept/vision'; nine for prioritisation including 'workflow'; nine for requirements including 'defining user stories'; and only one for feedback.

Cards also mentioned that the customer 'needs help to write stories' (mentioned once where proxy as customer); 'needs help to write acceptance tests' (mentioned once where proxy as customer); 'suggests how things work' (proxy as customer); 'transfer information about existing product' (very responsible customer). These are based on 14 project cards and again some included multiple responses.

Figure 7 What is other as customer responsible for?



3.5 Further comments

These are additional comments written onto the index cards. Some are in response to the question of whether the project was successful or not, e.g. 'good', 'average' etc.

- 'End-user competence wasn't clarified so sometimes overruled by the official representative of the stakeholder [i.e. not always clear that the 'customer' had the requisite authority]' [group of changing end-users in same room as developers]

- 'Problem: not enough time and interest' [developer on customer's site in the same room]
- 'Too weak separation between customer and developer – the consultant started considering developers as resources of theirs' [consultant was a proxy in the same room as the developers]
- ' proxy – in the corner office –when: everytime the customer has an issue what: raises priorities too high – out of control from the technical point of view. Typical in Italy (BAD)'
- 'who: leader of software development; where: in his office available by phone When: once a week, every week he promises to bring requirements but he had none. BAD Project cancelled without usable results'
- 'The project went well. Sometimes we answered our own questions' [2 end-users out of 3; 'Speak with one voice' communicated by email, phone calls]
- 'It was a good experience' [a real user, not on-site; calls once a month 'raise issues, ask for corrections']
- 'Customer was so excited about the project that he started developing with the team quickly. This was good at first (because he used the language of the developers). Later on problematic, because he lost the connection to his peers – customer identified himself more with developers than with his peers' [on-site end-user customer, joined development team].
- 'Good' [head of end-users, same company, different floor]
- 'Good' [project manager, not end-user, on-site – gave 60% of his time]
- 'Good – no external pressure' ['no real customer – product pushed for by the company. Team leader acted as customer]
- 'Not so successful' [virtual customer group for shrink-wrap product: engaged daily with the development team at standup meetings – and 'brings in stories and "decides" for which to go ("forced by management"?]
- 'Average' [business analysts, contractors for intra company product – on-site]
- 'very well' [whole product development team, none of them end-users but product conceptualiser; usability expert etc. 'did release planning on their own, were involved during iteration planning and accessible through the whole iteration']

3.6 Clustering criteria

Participants generally followed the outline suggested for capturing collaboration experiences, but when it came to clustering, they added some extra fields to capture factors they felt were important in characterising the customer relationship in an XP project. These were:

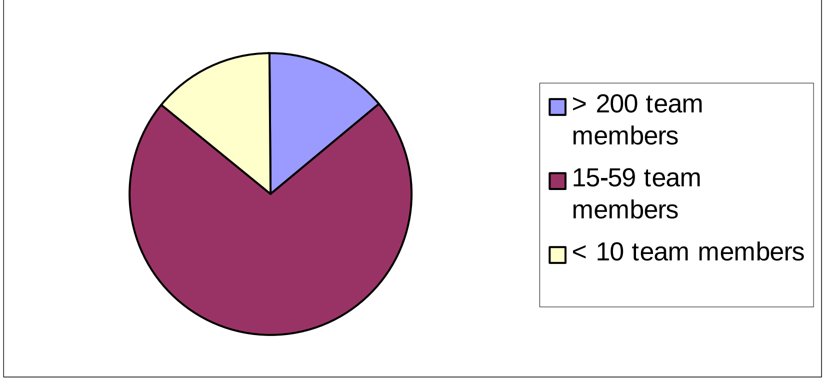
- Trust (which they later deleted as being too hard to determine at this point).
- Power (whether the customer actually had the power to determine requirements and priorities)
- Proxiness (was it a real customer?)
- Co-located or not
- Risk (how risky was the project)
- Ability to prioritise
- Project size
- Frequency of feedback (not the same as how often customers meet developers)

There was also general agreement that whether or not the project had been successful should be included on the cards.

3.6.1 Size

This was written in by the participants as being a significant discriminator between projects. It wasn't entirely clear what size meant ('everyone involved' someone said, but that is ambiguous), so we have just made distinctions into big (200 approx.) medium (15-50) and small (fewer than 10).
Of the 30 projects, under this classification, 2 were big, 10 medium, 17 small and 1 small – medium (the number of people involved fluctuated over time).

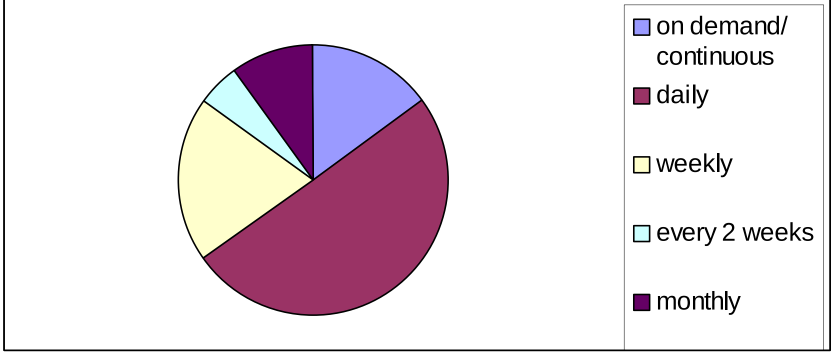
Figure 8. Size of project



3.6.2 Frequency of Feedback

Twenty cards had been labelled with 'frequency of feedback' information. Ten of the cards had no information about this. It is interesting to note that the attendees felt it significant to distinguish between developers meeting users, and developers getting feedback. For half of these projects, feedback was daily, while three projects had 'continuous' on 'on demand' feedback and four were weekly, one every two weeks and two every month (see Figure 9).

Figure 9. Frequency of Feedback



4. Group presentations

Results from the second task were captured as flip chart outputs and tape recordings of the discussion. There were 5 groups, divided according to the size of the project described on the index cards, except that the 'small' projects were divided between two groups. For each group we present the flip chart data and a summary of the verbal presentation.

4.1 Group 1 (large projects 50-200 people)

The customer's time is always limited and so we have the classic problem of balancing the demands of the day-to-day business of the customer and the project. The 'customer' should be a team, not one person because in the project they usually have different specialist areas. Using proxies is not a problem as long as they are good and experienced, and as long as they agree to represent the customers. If customers or proxies are overruled, this is a problem when there's a fight between the end users and the official end user representatives. For example one experience was reported where the requirements and design was done with the official end user representative but the acceptance tests were done with the real end users and there were differences between the two groups. If the project is very big it might be useful to divide it into sub-teams which are domain-specific with each sub-team having customers and developers.

Group 1 Flip chart data

Large Projects

- Customer knowledge resources are limited → day-to-day business vs project
- Customers must be a team → different domain knowledge
- Proxies are not a problem → as long as they're good and experienced
- Customers (or proxies) are often overruled → fight between end user and "official" end user representatives

Typically for "large projects"

- Requirements/design with "official" end users
- Acceptance tests with "real" end users

Domain-specific sub-team organisation

- With some service teams
- For products finding customers is more a problem

4.2 Group 2 (small projects avg size 15)

Two of the projects in this group were very successful, one was average and one was not so successful. It seems that successful projects are where the customer's view aligned with developers and everyone was clear about what they wanted to accomplish. For all of these cards, the customers were in the same room as the developers. That was perhaps because of the project size: between 5 and 20. This co-location and line of sight is interesting. Considering the pitfalls, of the cards in this group that weren't so good, the worst were technical misalignment and the 'goal owner' versus 'gold donor'[1] conflict. The pitfall there is that the customer doesn't have a single voice and there are arguments between the customer representatives.

Group 2 Flip chart data

Avg size = 15

Pitfalls

- Misalignment → goal owner vs gold owner
- No single customer voice
- Change of methodology mid-project

Avg size = 15

Success factors

- Alignment
- Location (→ frequency of availability)
  (might depend on project size?)
- All successful projects had "line of sight" distances

4.3 Group 3 (small avg 25)

Instant availability is a success factor, also the customer should take hard decisions and take high responsibility for project success. To steer the project, they have to have project vision. For pitfalls, in two projects, mentoring was necessary to establish customer practices like writing stories, accepting their responsibility, accepting the estimate of the developers and taking part in the acceptance tests. There was one instance where the user was getting more and more disconnected from the real domain experts. Misalignment between 'goal owner' and 'gold donor' was another problem.

[1] The 'goal owner' is the person or people who champion the goal of the project. The 'gold donor' is the person or people who pay for the project.

Group 3 Flip chart data

Success factors

- Instant availability
- Hard trade-off decisions
- Project vision

Pitfalls

- Mentoring for customer practices needed
- Disconnectedness to "real" domain experts
- Losing the responsibility for the customer role
- Lack of acceptance tests
- No alignment between gold owner and goal donor

4.4 Group 4 (small projects 25-50)

Starting with the pitfalls of small projects. One of the pitfalls is if the customer is unsuitable for collaboration; the customer has to have a voice and not be opposed to the project. Another pitfall is that the customer disturbs the developers. It is easy for him to say 'well, I have a problem with Excel can you help me with that?' or 'this other application that your company developed for us, we have a problem with that'. That disturbs the developers. Another problem is that the customer can see too much when located on-site. For example, he/she sees the developers working and hears them chatting, which is likely to include some good things and some bad things. The customer is also less productive for his/her own daily tasks. While he/she is away, work piles up and when she/he goes back there is a lot of work to be done. Another pitfall is if the customer directs the developer, using him/her for his own resources.

For the positive points, there are wonderful acceptance tests. Instead of making your own test sets and your own scenarios to test the application the customer does that and also sees what the potential problems are in practice. The developers get direct feedback, for example, on a design decision. The customer is more talkative, and directly or indirectly he gives you leads for more work for your company. Also, developers are more focused on the project and less likely to be distracted by visitors or other issues.

Group 4 Flip chart data

Small projects
Frequent communication

Positive

- Wonderful acceptance test
- Direct feedbacks
- Leads for more work
- Developers are more focused on the project

Small projects
Frequent communication

Negative/pitfalls

- Customer not suited for collaboration
- Customer disturbs developer(s)
- Customer sees "too much" (confidentiality)
- Customer is less productive for his "own" daily tasks
- Customer directs the developer: uses the developer as his own resource

4.5 Group 5

In this group, they talked about when is XP not necessary or even when is XP not so helpful. The critical success factor of having an XP project is that the customer cares about the project vision so everyone knows the core requirement. An on-site customer might be considered harmful, both for developers and for the customers. Daily meetings for communication should then lead to some separation time for developers and customers.

The customers have their daily work and they do not want to be bothered about steering this project on a daily basis. Meeting overhead needs to be monitored. One very important issue is that there should be a hot-line so that communication is immediate both from the customer point of view and for the developer point of view. If the developer considers an issue to be important then they should be able to get an answer quite quickly. There should be a customer that can steer the project, that knows about the main goals, that can identify the main borders and that all the rest is taken care of out of the project team. On the other hand the project team should understand the customer, should understand the customer's needs and should be able to think like they are the customer.

Group 5 Flip chart data

Small project size ≥ weekly meetings

Critical Success Factors:

- Customer cares about core requirements, not the details
- Daily comm considered harmful → over-controlling
  → waste of time
  → meeting -overhead
- Need for

  Regular status meetings

  Communication on demand (hot-line)

  Customer that can steer the project, identify main target

  Project team that can understand customer needs

5. Summary

The session allowed attendees to exchange experience, and to discuss some of the key factors in successful XP projects. From the analysis above, it appears that there is variety in terms of the approach taken to customer collaboration in XP projects. Of our projects, half of them involved end users as the customers and half were 'proxy' users. The customer was on-site with the developers for fewer than half the projects reported on. In most cases, customers were responsible for the core activities prescribed in the popular XP books, although very few were recognised as being responsible for steering the product, and surprisingly few were responsible for feedback (especially where the customer was not an end-user).

Success factors that were common across project size included alignment of customer and developer objectives, and appropriate levels of communication, i.e. not too much and not too little. Pitfalls appeared to be the converse of the success factors: where there was no alignment between developers and customers, or even between customers, and where communication failed.

References

Beck, K. (2000) *eXtreme Programming Explained: embrace change,* Addison-Wesley, San Francisco.

Gould, J. D. and Lewis, C. H. (1985) Designing for usability: key principles and what designers think, *Communications of the ACM,* **28**, (3)**,** 300-311.

Heinbokel, T., Sonnentag, S., Frese, M., Stolte, W. and Brodbeck, F. C. (1996) Don't underestimate the problems of user centredness in software development projects - there are many!, *Behaviour and Information Technology,* **15**, (4)**,** 226-236.

Keil, M. and Carmel, E. (1995) Customer-Developer Links in Software Development, *Communications of the ACM,* **38**, (5)**,** 33 - 44.

Kotonya, G. and Sommerville, I. (1998) *Requirements Engineering: processes and techniques,* John Wiley & Sons, Chichester.

Robertson, S. and Robertson, J. (1999) *Mastering the Requirements Process,* Addison-Wesley, Boston.

Taylor, A. (2000) IT projects: sink or swim, *The Computer Bulletin,* 24-26.

TCRE02 (2002) *Proceedings of the International Workshop on Time Constrained Requirements Engineering*, Essen, Germany,