Extreme Support

Rachel Davies & Colville Wemyss
Connextra
Studio 312 Highgate Studios
53-79 Highgate Road
London, NW5 1TL UK
+44 20 7692 9898
support@connextra.com

ABSTRACT
This subject of this report is Client Support[1] in an agile software development process, specifically in the context of XP, based on the experience of the team at Connextra.

In traditional software organizations, Client Support is normally provided by a dedicated team that liaises with end users to provide help on the basis of their product knowledge base and feeds bug reports into the Development organization.

However, in our experience this strategy does not port well to organizations using agile software development methods, such as XP.

The XP team at Connextra has evolved a new collaborative strategy for handling Client Support over the last two years and that is the subject of this experience report.

Keywords
XP, Support, Maintenance

1 INTRODUCTION

XP is a development process tuned for fast-moving business requirements: it is particularly effective in new product domains because it has a lower cost of change. XP teams are often new start-ups, and there's no need for client support until you have released a software build that is deployed to a significant user base. In other software development processes, e.g. RUP, there's a deployment phase which is not really covered in any detail by books published on XP (at the time of writing).

This topic is touched on in "Questioning XP" [1] where McBreen asks "Is a Handoff to Maintenance Useful?" and concludes that XP teams assume a near-continuous stream of new requirements, the original development team may be scaled back but continue to support the application indefinitely.

Background

This paper is a report on the experience of an XP team at the company, Connextra, in providing support to clients in external organizations over the period of the past two years. To give the reader some background, Connextra was founded on XP principles mid-1999 and has now become a market leader in providing tools to generate web advertising with dynamic content.

The organization is fairly small, divided broadly into two teams, a Customer team (responsible for strategy, marketing and sales) and a Development team (responsible for technical architecture, software development and deployment). Team size has fluctuated but at present there are four people in the Customer team and ten people in the Development team (the majority are Java developers with one web designer, one system administrator and one person dedicated to Client Support).

The team at Connextra has maintained a commercial customer base for the product over the last two years in parallel with product development work. There are stories in the product roadmap for at least another year. The product is still evolving and yet the commercial reality is that a good standard of Client Support must be maintained in order to develop our customer base.

As early adopters of XP, Connextra were amongst the first to grapple with the challenge of finding an appropriate solution to Client Support in the context of XP. In the absence of documented XP practice, the team at Connextra used the values of XP as their guide. Simple solutions were sought, refactored using feedback from the team in retrospective meetings to make informed choices, with a rationale communicated across the whole team.

2 DEVELOPMENT TEAM PROVIDE SUPPORT

Where the demand for client support is low, XP teams may elect to absorb the Client Support role into the Development team. This was the approach taken by Connextra in the first few months of deployment. An XP team uses Test Driven Design and as such has good code test coverage, which leads to a low defect rate so this means there should be less need for client support caused by "bugs". Support requests may however indicate missing test scenarios.

At Connextra, we found that providing Client Support as part of the Development iteration caused the following problems.

---

[1] This paper uses the term "Client Support" rather than "Customer Support" to avoid confusion with XP "Customer" role.

Interruptions
When working on a complex task, programmers immerse themselves into a problem by building mental models of the problem. Interruptions disrupt developers' "Flow" meaning lost time in rebuilding mental models. In "Peopleware" [3] each interruption to a person working on their own cost 15 minutes re-immersion time. The authors of this paper are not aware of any published figures for pair programming re-immersion time but it is likely to be of a similar order. The open-plan workspace required for pair programming may also mean that several pairs are distracted from their planned tasks by a single interruption. Indeed, Collective Ownership can exacerbate this problem as no module has an identifiable owner, so several developers may need to be asked about a feature before someone who worked on that story is found.

Planning
Client Support is typically piecemeal and reactive in nature, likely to be to according to the client's schedule and as such is sporadic, so not amenable to planning. If a Development team undertakes Client Support, they must leave sufficient slack in their plan to allow for a certain amount of contingency resource for Client Support.

XP teams use Velocity to scope iterations, so unless there is a constant level of Client Support work, a fluctuating number of support requests can impact the XP planning process.

Continuity
If Client Support is undertaken by the Development team, it may be hard to maintain continuity from the Client's perspective; information "falls between the cracks" as different developers handle Client Support requests and this can cause frustration as clients have to go over the same ground.

3 SEPARATE CLIENT SUPPORT TEAM

Using a dedicated Client Support team gives the client more continuity; they have a known point of contact in the team. A Client Support team can absorb interruptions and provide resource buffer.

At Connextra, we found that following the traditional approach, making Client Support the responsibility of a dedicated Client Support team, did resolve problems related to continuity and impact on iteration plan but not the issue of interruptions, now coming from Client Support.

Tacit Knowledge
The Agile Manifesto [2] declares that working software is valued over comprehensive documentation. Agile teams prefer direct face-to-face communication rather than indirect forms of communication, such as documents. Direct communication is richer and allows the team to save time on maintaining documents by relying instead on accumulating tacit knowledge.

An XP team puts more effort into writing code that is intention-revealing so that comments are rarely needed by developers and knowledge about the evolving software design is shared by using Pair Programming. Information is exchanged verbally in the Planning Game and daily stand-up meetings, making use of an Onsite Customer during the day for questions.

The Client Support team needs access to knowledge about the software design held by the Development team and this presents a challenge to Client Support, as to do this they must be "in the loop". With limited knowledge of the evolving software design in the Client Support team, consultation with Development is necessary in order to give a client a rough estimate for a new feature or to determine the appropriate fix for a minor bug.

4 EVOLVED CLIENT SUPPORT PROCESS

For the past year, the team at Connextra has operated the following Client Support process.

One person is allocated permanently to a Client Support role and each day a developer pairs with that person to provide technical assistance. We call this developer "exposed": this means interruptible for Customer and Client Support questions. The exposed developer may need to ask another pair for help but (from paired work on the code base) they will have a better understanding of who is the best person to ask. All developers take on this role in turn.

Each day in Client Support is effectively a micro-iteration. Clients request changes to current software behavior, these requests may be resolved by data changes or small coding changes and deployment of those changes to live web servers. Each request received by phone or email is entered into a tracking database, allocated a ticket number and then written up on an index card and pinned on a pin-board.

The Client Support pair holds a stand-up meeting with the Customer team in front of this board every morning, directly after the Development stand-up meeting. The purpose of this meeting is to determine commercial priorities for the day, although these may reprioritized as the day unfolds[2].

During the day the Client Support pair is responsible for performing "triage" on any new support request received.

- Some requests may need immediate action, typically where software is used in some unexpected scenario (indicating a missing acceptance test) and the service provided to the client is significantly impacted.
- Other requests may be placed at the end of the Client Support work queue and reviewed in the standup the next day.

[2] Five numbered Velcro arrows are used for rapid adjustment (we move the arrows not the cards), so the Customer team can see the top of the priority stack.

- Any requests that require more than half a day of paired development time are treated as new story candidates for the next development iteration to be prioritized in the Planning Game. In this case, the Client Support pair may be asked to give a rough estimate of the work likely to be involved, to help them to manage the client expectations.

5 EVOLVING OUR PROCESS

Following advice gleaned from the XP2001 conference [4], the team at Connextra started to hold a Retrospective at the end of each iteration. Using this forum, the team has raised issues related to client support; for example, uncontrolled interruption levels and concern about continuity gaps. Solutions were proposed in these retrospective meetings and then tried out in subsequent iterations.

Among the other ideas we tried were:

- The use of a "Lone Ranger": some days we have an odd number of developers in the office, so a person who has no one to pair with could be dedicated to support tasks.
- An "Interruptible Pair" from the Development team, placing their names on index cards at the entry point to the Pair Programming area of the office. This Interruptible Pair provided the first point of contact for questions from clients and customers.

However, the Interruptible Pair and the Lone Ranger solutions suffered from the loss of continuity problem between client requests (described above).

6 CONCLUSION

The team at Connextra found benefit in using Retrospectives to improve our Client Support based on team feedback. The team has also used this approach to improve our development and business processes as some support requests can be traced back to vague Customer stories or inadequate Acceptance tests.

Our evolved Client Support process fits in with XP and Agile values. The fact that developers have to "eat their own dog food" brings valuable feedback into Development team. The collaborative approach means Communication has improved throughout the team. The Client Support process is specifically set up to handle change methodically without compromising the business.

The authors are aware that the team at Connextra is relatively small and so collaborative Client Support may not scale to larger organizations. However, the more general idea that both the support team and developers benefit from developers regularly taking a "tour of duty" in a support role is not restricted to small organizations.

7 INFORMATION AND QUESTIONS

For more information, please contact: rachel.davies@tiscali.co.uk

ACKNOWLEDGEMENTS

The paper would not have been possible without the experience of working with the team at Connextra under the guidance of John Nolan and Tim Mackinnon.

This document has been adapted from a submission accepted for XP Agile Universe 2003, the authors are unable to present the paper there but the reviewers comments were taken into account and their contribution is gratefully acknowledged.

REFERENCES

1. McBreen, P: Questioning XP. Addison-Wesley 2002
2. Agile Manifesto: on-line at http://agilemanifesto.org 2001.
3. DeMarco,T & Lister,T: Peopleware Productive Projects and Teams. Dorset House Publishing 1999.

4. Collins, C & Miller, R: "Adaptation: XP Style". XP2001