

# Interaction Designers on eXtreme Programming Teams: Two Case Studies from the *Real World*

Jennifer Ferreira<sup>1</sup>, James Noble<sup>1,2</sup> and Robert Biddle<sup>3</sup>

<sup>1</sup> Victoria University of Wellington, New Zealand

<sup>2</sup> Microsoft Research Cambridge

{jennifer,kjx}@mcs.vuw.ac.nz

<sup>3</sup> Carleton University, Ottawa, Canada

robert\_biddle@carleton.ca

**Abstract.** The interaction designer role is not an acknowledged role on the core eXtreme Programming (XP) team and XP has no explicit process for dealing with interaction design. We interviewed interaction designers and other team members on two real-world XP teams and in this paper we report on how they combined interaction design activities with XP. Initial results show that having interaction designers on the team, resulted in a workflow that differed from the pure XP process in a significant way: up-front design for user interaction, as opposed to up-front code design, *was* considered necessary. The iterative nature of XP development required that the interaction designers have continual involvement with the development of the product, which inevitably influenced the nature of the relationship between the interaction designers and the developers.

**Keywords:** Software Engineering, System Usability, eXtreme Programming, Interaction Design

## 1 Introduction

Extreme Programming (XP) [1] is the most widely adopted agile method today [2], however, it has no explicit process for dealing with interaction design. Studies have shown that companies that produce software with poor usability lose money [3–5]. Anderson connects usability directly with value: “Value is perceived through usage. Without usability there is no value.” [6]. Interaction designers can help the end user perceive that value by enhancing the usability of the product.

We conducted interviews with interaction designers<sup>4</sup> and other team members on two real-world XP teams to learn about the role of the interaction designer and how interaction design activities are integrated with XP. Our hypothesis was that the role of the interaction designer would differ from that of interaction designers on traditional software development teams and would also be key in the success of the integration of

---

<sup>4</sup> Although the interviewees used the terms *interaction designer* and *user interface designer* interchangeably, we use the term *interaction designer* to refer to the member of the development team, whose main responsibility it is to design the user experience and the user interface. The team members involved in mainly coding activities, are referred to as the *developers*.

interaction design and XP. Our preliminary results show that XP teams have different approaches to combining user interaction activities with their development process, but wholeheartedly agree on the valuable contribution of the interaction designers to the development of their product. They see a real need for usability specialists, in the form of interaction designers, to be involved with development because experience has shown that following the XP rules is not enough to ensure usable software. Our research also shows that with the interaction designer role added to the team, the workflow differed from the pure XP process in a significant way: up-front design for user interaction, as opposed to up-front code design, *was* considered necessary. Further, the iterative nature of XP development required the interaction designers to be continually involved with the development of the product, which influenced the nature of the relationship between the interaction designers and the developers.

In the next section we explain our research method. The teams we interviewed are introduced in section 3 and the interaction designer role is explored in section 4. After a brief discussion of the related work in section 5, we present our conclusions in section 6.

## 2 Research Method

The data for this paper was obtained from semi-structured in-depth one-on-one interviews with four team members from two different software companies: Greenback Inc. based in the United States and Emerald Inc. in Ireland.<sup>5</sup> The two main objectives of these interviews were to understand the process and practices relating to interaction design on XP projects and to learn about the interaction designer role in XP teams. The interviewer conducted the interviews with team members from Greenback Inc. on their premises. The interviews with the team members from Emerald Inc. were conducted using phone conferencing facilities. The interviews were voice recorded and transcribed in detail. All persons interviewed were asked to validate the transcriptions, as well as the interpreted findings. We present those findings here and quote the interviewees as illustration.

The interview transcriptions were analysed using the method known as open coding [7]. This method is the first step in grounded theory analysis [8] and is used to identify the different recurring categories present in the data. The next step will be to perform axial coding, where the relationships between the categories are established. In the course of further analysis, more interview data will be incorporated to eventually produce a theory grounded in the data.

## 3 Team Profiles

Based in the United States, Greenback Inc. is a company that develops and markets web-based software to support IT managers and development professionals. Team Liberty of Greenback Inc. is an XP team and includes ten engineers and one user interface designer/product manager. At the time of the interviews, Team Liberty was working on redesigning and enhancing one of its products. Their User Interface Designer/Product Manager described the project as follows:

---

<sup>5</sup> All names have been changed in order to preserve confidentiality.

“There are several features we added and several things that we wanted to do with the product. And one thing we noticed was that performance was really bad, it was built upon a really terrible code base. It was just all hacked together. User interface was terrible and, you know, the user interactions were very cumbersome. So we decided from that generation of product that we were gonna rewrite and start from scratch. We took everything that we wanted out of the old product and built it the way we really wanted the product to end up. [...] During this process we adopted the XP methodology.”  
— *User Interface Designer/Product Manager, Team Liberty*

The second team, Team Cláirseach, is employed by Emerald Inc., based in Ireland. This company develops and sells software to support wealth management. Team Cláirseach is also an XP team and includes four engineers, one domain expert/on-site customer and two interaction designers. Their Project Manager and one of the interaction designers described their project and its status at the time of the interviews:

“We’ve been focusing on, sort of, single-user client systems.” — *Project Manager, Team Cláirseach*

“There are smaller projects as well and we have our bigger, our over-riding kind of application building, which is our wealth planner application, which we’ve been working on for two years and we have our first customer for that now and we’re releasing that in a few weeks. And we’re certain of a way through developing. It’s quite big, so it’s an ongoing project [...]” — *Interaction Designer, Team Cláirseach*

Team Liberty provided the Engineering Manager and User Interface Designer/Product Manager for interviews and Team Cláirseach provided the Project Manager and Interaction Designer. The Engineering Manager and Project Manager were both from a programming background, whereas the User Interface Designer/Product Manager and Interaction Designer were both experienced and qualified user interaction specialists.

## 4 eXtreme Interaction Designers

During the open coding of the interviews, we became aware that the interaction designer role presents unique challenges in the world of XP. In this section we take a look at how the various up-front activities formed part of the development process and the nature of the interaction designer/developer relationship.

### 4.1 Up-Front Activities

In XP the development process is an evolutionary process, with little or no significant design up front. Beck states that “it is better to do a simple thing today...than to do a more complicated thing today that may never be used anyway,” [1] that is, “Concentrate on what is scheduled for today only.” [2] Among practitioners, this is known as the YAGNI (You Ain’t Gonna Need It) principle and warns against adding features to the product that are not required in the current development iteration. Therefore, to avoid waste and to remain responsive to changing requirements, XP discourages up-front code design. While neither Team Liberty nor Team Cláirseach performed up-front

**code design**, both teams found a significant amount of **user interaction** design, after the requirements gathering process and before implementation begins, to be crucial for usability. Both teams created *personas*<sup>6</sup> to represent the types of users of their systems. The following quotes illustrate the value of the personas:

“[...] as we created the user stories all written from these personas, what that allowed me to do was figure out who was trying to do what [...] and it just helps in the design process.” — *User Interface Designer/Product Manager, Team Liberty*

“[...] that [personas] was for the UI, but also for the user stories, one of the most important things to do. [...]” — *Engineering Manager, Team Liberty*

“[...] We design personas up front for our projects and we identify what their goals are in using the product. We have to make sure that when we’re testing the product we meet all their goals.” — *Interaction Designer, Team Clairseach*

Team Liberty then created a navigation model and a style guide in order to set the interaction standards for the future user interface and to provide some consistency in its behaviour and appearance:

“I’d try to work things out. Just create some upfront consistency, like, what do buttons look like, where are they placed, what do tables look like, how do users interact with tables, what do forms look like, how do you get from a table to a form and then back to the table – basic interaction models. So, kind of like a style guide. [...] having that upfront consistency in designing the behaviour into it, regardless of what the application is, I think, is one way of mitigating risk. And then having a rough, high-level navigation model, I think, is probably a good idea too.” — *User Interface Designer/Product Manager, Team Liberty*

“[...] it’s hard to have a holistic view of the application when you haven’t followed out all its framework: How am I gonna handle breadcrumbs, how am I gonna handle main navigation, how am I gonna handle user profile stuff. [...] So I created a really basic framework in the navigation model [...]” — *User Interface Designer/Product Manager, Team Liberty*

Team Liberty refined and implemented the detailed user interface designs incrementally during the XP development iterations. By contrast, Team Clairseach completed the design of the bulk of the user interface before implementation began:

“Before it gets into development, the user interface is more or less 90% defined. So there wouldn’t be that many changes once it goes into the development iterations [...] because we put so much effort into the [up-front] design.” — *Interaction Designer, Team Clairseach*

## 4.2 Continuous Involvement

In popular software engineering processes, such as the Rational Unified Process (RUP) [10], the interaction designer is responsible for the complete design of the user interface early on in the development process. Then the design is handed to the developers

---

<sup>6</sup> A well-known technique introduced by Alan Cooper [9]

who implement the design, with little or no further involvement from the interaction designers. In the XP teams interviewed, this was not the case. The interaction designers on these XP teams were involved in the up-front activities discussed in section 4.1, but from that point on, as the user interface was implemented during the development iterations, the interaction designer was constantly communicating with the developers regarding changes, testing and clearing up any questions the developers may have had about the design. For the designers it was possible to receive immediate feedback from the developers when something could not be implemented, and the developers could be informed by the designers straight away when there were additions or changes to the user interface.

“[...] the designers test it [the user interface] on a day to day basis, give feedback back to the development team to ensure if anything was missing, that we’d write a card for it and it will be captured.” — *Interaction Designer, Team Cláirseach*

“[...] sometimes even during development people realize, ‘Oh this and this doesn’t work,’ and then they go to the User Interface Designer/Product Manager.” — *Engineering Manager, Team Liberty*

With this continuous involvement the interaction designer could ensure that the developers maintain consistency in the user interface, as specified either by the style-guide or the mock-up.

The Engineering Manager on Team Cláirseach related an incident where the communication between the designers and the developers had broken down, leading to major bottlenecks in the development process:

“[In] London a guy had serious problems [...] he had an interaction expert guy who [would] not talk to the programmers and start changing stuff [...] There was very little communication between the different blocks and it caused serious delays and bottlenecks [...] and not knowing what he’s doing he’s causing them more work.” — *Engineering Manager, Team Cláirseach*

### 4.3 Designer-Developer Interaction

The relationship the interaction designers have with the developers on the XP team is a direct consequence of their continuous involvement, as discussed in the previous section.

We observed two different approaches to the interaction designer/developer relationship on the XP teams and that difference is clearly marked by the following two quotes. Within Team Liberty the developers were free to give their User Interface Designer/Product Manager feedback on user interface design issues:

“[...] the Product Manager comes to the meeting and says, ‘Ok, here’s how we do this and this feature,’ and there’s ten engineers sitting there and saying, ‘Look, nobody works like this. What are you doing?’ [...] but it’s a fantastic thing; you have these sometimes very opinionated discussions up front and sometimes even during development.” — *Engineering Manager, Team Liberty*

Whereas, in Team Cláirseach, developers did not give feedback on user interface design issues:

“Pretty much design-wise, whatever the interaction designers say goes.” — *Engineering Manager, Team Cláirseach*

Team Liberty valued lively discussions and debate and enjoyed sharing knowledge of their domains and the overall product. They believed this helped them come up with the best solution. Interaction design in this team was clearly seen as a collaborative activity. Unfortunately, the Engineering Manager admitted that it was sometimes difficult to convince the developers to implement a screen that they did not like or did not agree with. This was largely due to the fact that the developers did not take into account how other types of users could use their software. By contrast, their user interface designer is trained to take all the different types of users into account:

“[...] there’s ten engineers sitting there and saying ‘Look, nobody works like this. What are you doing? It’s like, nobody’s doing this,’ and so sometimes it’s really, really hard for him [*User Interface Designer/Product Manager*] to tell them, ‘Look, you might not work like this but I think outside, people who use our application, they will work like this, they will appreciate this feature.’” — *Engineering Manager, Team Liberty*

The designer/developer relationship in Team Cláirseach, is based around respect and trust. The designers show their respect for the developers by not suggesting coding improvements and the developers show their respect for the designers by not suggesting user interface improvements:

“I have to say in our team here, each of us have great respect for each other’s work. We, as designers, have great respect for what the developers do and the developers have great respect for us. Even with great communication going, we’d never assume to make a suggestion about something we don’t know very much about.” — *Interaction Designer, Team Cláirseach*

Consequently, the interaction designers on Team Cláirseach have complete control of the user interface. Further, designers and developers trust each other that each will come up with the best solution for their domain and when issues do arise, both sides make trade-offs until both are happy. This team viewed their approach as a very efficient way of developing the product, as designers and developers do not waste time and effort on things they are not good at. Their view was that their work in their own domain is better when they do not have to think outside the scope of their domains:

“If I don’t have to worry about UI concerns, I can get more work done and get it done better. The UI doesn’t naturally fall inside my domain, it isn’t something I naturally do well.” — *Project Manager, Team Cláirseach*

Although not explicitly mentioned by Team Cláirseach, one disadvantage for this team may be that knowledge is not shared among team members, which would make it difficult for team members to learn from each other.

#### **4.4 Valued Role**

Both teams agreed that having interaction designers on the XP team was vital for enabling good interaction design. In their opinion, having the developers doing interaction design is not ideal:

“[...] you need someone who really has a very good idea about the really high level UI design.” — *Engineering Manager, Team Liberty*

“I don’t think the user interface designers should be engineers.” — *User Interface Designer/Product Manager, Team Liberty*

“I can’t imagine why a developer would be designing screens because their training lies in a whole different area to me, and their skill set lies in a different area [...]” — *Interaction Designer, Team Cláirseach*

## 5 Related Work

The way in which interaction design and agile development should work together has been discussed surprisingly little. One important early exception was the debate between Kent Beck and Alan Cooper [11]. This debate explicitly addressed the issue of when interaction design should occur relative to software development. Jeff Patton describes in several papers and tutorials how interaction design and agile development can work together by using a process where interaction design iterations fit into the iterative structure of agile development [12, 13]. Hodgetts presented an experience report about integrating User Experience Design into agile processes [14] and McInerney and Maurer investigated the role of User-Centered Designers on agile teams [15]. The study that is most closely related to the research presented in this paper is a case study by Rogers, Sharp and Preece [16]. Although their findings are specific to designing a web-based interface, they focus on the work of the graphic designer and how she relates to the XP team — a similar approach to that taken in our research. There are, however, very few independent studies of the interaction designer’s role in XP teams: a gap this research hopes to begin to fill.

## 6 Conclusion

We set out to investigate the processes and practices of interaction design of two real-world XP teams and uncovered some unique characteristics of the interaction designer role. The approach both teams interviewed have taken, differed in significant ways from the pure XP process, in that both teams performed significant up-front user interaction design — producing user interface mock-ups and style and navigation guides. The iterative nature of XP development, required continual involvement from the interaction designer, which differs from traditional interaction designer roles. Both teams acknowledged the value of having interaction designers on board and believed they helped create a better, more usable product.

## References

1. Beck, K.: eXtreme Programming Explained: Embrace Change. Addison Wesley (2000)
2. Pettichord, B.: Deconstructing GUI Test Automation. STQE Magazine (January/February 2003) 28–32

3. Marcus, A.: Return on Investment for Usable User Interface Design: Examples and Statistics. Aaron Marcus and Associates, Inc. Whitepaper (2002) Published at URL: [www.amanda.com/resources/ROI/AMA\\_ROIWhitePaper\\_28Feb02.pdf](http://www.amanda.com/resources/ROI/AMA_ROIWhitePaper_28Feb02.pdf).
4. CC Pace: Usability and User Interface Design in XP. Available at URL: [www.ccpaace.com/Resources/documents/UsabilityinXP.pdf](http://www.ccpaace.com/Resources/documents/UsabilityinXP.pdf) (Last accessed 15 December 2006.)
5. Nielsen Norman Group: Usability Return on Investment. Available at URL: <http://www.nngroup.com/reports/roi/> (Last Accessed 15 December 2006.)
6. Anderson, D.J.: Scheduling UI Design with Critical Chain Project Management. Published at URL: [www.agilemanagement.net/Articles/Papers/Scheduling\\_UI\\_Design\\_v1\\_5.pdf](http://www.agilemanagement.net/Articles/Papers/Scheduling_UI_Design_v1_5.pdf) (Revised October 2004. Last Accessed 15 December 2006)
7. Strauss, A., Corbin, C.: Basics of qualitative research: techniques and procedures for developing grounded theory. Thousand Oaks (California): Sage Publications (1998)
8. Glaser, B.G., Strauss, A.L.: The Discovery of Grounded Theory: Strategies for qualitative research. Aldine de Gruyter, Hawthorne, NY, USA (1967)
9. Cooper, A.: The Inmates are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity. (Indianapolis): SAMS (1999)
10. International Business Machines Staff: Rational Unified Process: Best Practices for Software Development Teams. Rational Software Corporation Whitepaper (2003) Published at URL: <http://www-128.ibm.com/developerworks/rational/library/253.html>. Updated July 2005.
11. Fawcett, E.: Extreme programming vs. interaction design. FTP Online (2002)
12. Patton, J.: Improving on Agility: Adding Usage-Centered Design to a Typical Agile Software Development Environment. In: ForUse2003: Proceedings of the Second International Conference on Usage-Centered Design. (Portsmouth, NH, USA, 2003)
13. Patton, J.: Usage-Centered Design in Extreme Programming and Agile Software Development Environments. In: *Tutorial* at ForUse2003: Proceedings of the Second International Conference on Usage-Centered Design. (Portsmouth, NH, USA, 2003)
14. Hodgetts, P.: Experiences integrating sophisticated user experience design practices into agile processes. In: Proceedings of Agile 2005 Conference, Denver, CO, IEEE Computer Society (2005) 235–242
15. McNerney, P., Maurer, F.: UCD in agile projects: dream team or odd couple? *interactions* **12**(6) (2005) 19–23
16. Rogers, Y., Sharp, H., Preece, J.: Using XP to Develop Context-Sensitive Adverts for the Web. In: Companion website for the book *Interaction Design: Beyond Human-Computer Interaction*, at URL: [http://www.id-book.com/casestudy\\_xp.htm](http://www.id-book.com/casestudy_xp.htm), Wiley (Last Accessed 15 December 2006)