

XP – Call in the Social Workers

Tim Mackinnon

Connextra Ltd., London, England
tim.mackinnon@pobox.com

Abstract. XP has the potential to allow a team to work on a project for a long period of time, potentially several years. At Connextra we have been carefully following the suggested XP practices for more than 3 years, and along with additional techniques such as team retrospectives and gold cards, we have proven that sustainable development is a reality. This paper outlines the history the team, the introduction and refinements of team retrospectives and an experiment in a cross discipline exchange with a qualified social worker.

Introduction

Connextra was founded in the summer of 1999 with 3 employees and the aim to develop a desktop product that shows contextual information relevant to what a user is viewing on their computer. After creating some initial prototypes, two of the founders began writing the application using some of the suggested Extreme Programming (XP) [1] practices. It quickly became apparent that a larger team would be required to complete the application, and full XP was chosen as a strategic methodology for accomplishing this work. The author, who had prior XP experience, was encouraged to join the project to set up the XP team and develop the product.

Growing The Team

With the original founders sorting out details for a growing company, it was imperative to locate some additional team members with an aptitude for XP. Contacts proved important in this search, and a previous student intern was tracked down along with another recommended student graduate. Both new team members demonstrated an ability to learn new techniques along with a well grounded programming knowledge based on object-oriented (OO) programming. With this team we were able to work with two rotating pairs and hold a planning game every 3 weeks. At this time we also invented a new testing technique called Mock Objects [6], which we used as a means of successfully applying test first programming to our problem domain.

While XP does not formally address hiring practices, we quickly noticed that collective ownership meant we were reliably able to establish whether a potential employee was a useful addition to the team. The pattern we established began with a 30-minute telephone interview and asked a few simple Java, HTML and OO

2 Tim Mackinnon

questions. If successful, the next step was to invite the candidate for a face-to-face meeting at our offices where we discussed more general programming issues and talked about Extreme Programming. If the interviewee then showed a potential for problem solving and had a curiosity about XP, we then invited them to spend half a day with our development team pair programming. As this request can be quite intimidating, it was clearly explained that they would not be expected to immediately understand our code base or use our development tools to their full extent, however they would be expected to interact with their partner and use them as mentor to work on simple problems.

At our morning standup, if there was an interview, we all agreed on a good development task and roughly which team members would rotate through the course of the session. As our code base is very decoupled (due to our Mock Object discipline) it was possible to work on a small task without having to know every detail about how other parts of the system functioned. As the candidate built up tests with a partner we switched them to a new partner and looked for an ability to explain what they were trying to accomplish and what they were currently working on. At the end of the session the participating team members would have a quick discussion about the candidates' ability to write simple tests, and keep track of progress. We also looked for a capacity to learn something new during the session (typically the usage of our tools or programming techniques) as well as a sense of humor or a passion for some subject. Using this simple criterion we have successfully maintained a team of the original members. Furthermore, we have also had several candidates that have released production code in their interview, meaning that when they were hired it was actually their second day on the job.

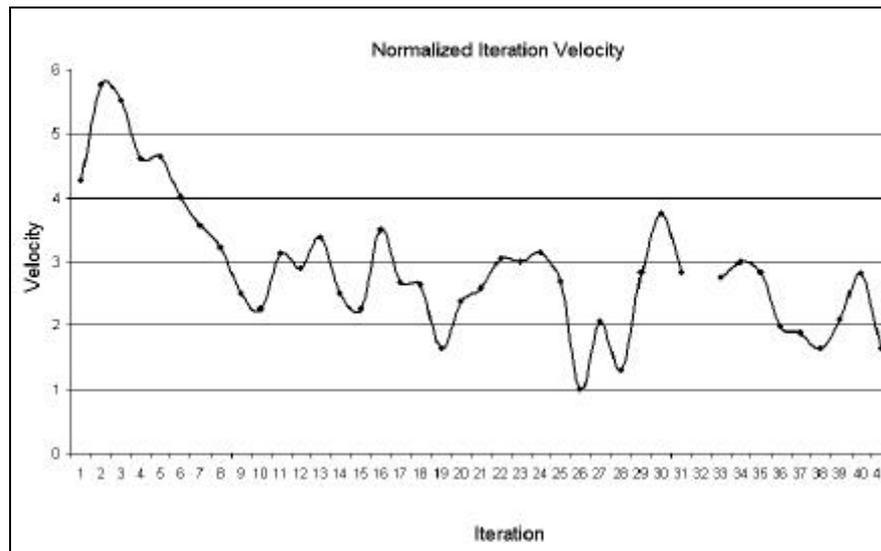
Iteration History

With a running team in place, during the course of 3 and half years we have measured (and continue to measure) our velocity for each iteration (fig. 1), and used that number to drive each subsequent planning game. Initially we used an iteration length of 3 weeks, however we later adjusted this to 1-week iterations. For consistency and to compare results between iterations, fig. 1 shows a normalized velocity for a three week total based on one pair of developers. The early iterations had quite a high velocity when we didn't have any issues of support or hot staging to live servers.

Although we continued to monitor our velocity and used it as a means of planning subsequent iterations, we didn't initially use it as a way of monitoring team happiness. We viewed the numbers as a way of accommodating technical risk and reliable delivery (via yesterday's weather). Of course at each planning game we had to explain the change in numbers to our users, and as a team we would informally discuss ways of improving or maintaining our process. After working like this for over a year, we found that everyone was finding the work a little monotonous. Thus at iteration 17, we introduced the concept of Gold Cards [5] to enable developers to have planned time to research new technologies and tools. However, while the feedback we were getting was that gold cards were helping improve moral, and we were

continuing to deliver new functionality, there was still a problem that seemed to be eating away at the morale fiber of the team.

Fig. 1. Normalized Velocity (for a single pair) recorded over 41, 3-week iterations.



Although the velocity appears to jump around, in general we have maintained a figure of 2.5 to 3 per pair (a number which includes gold card research and support tasks)

Dissatisfaction Tracking

Unable to directly identify the cause for unrest, we decided at iteration 18 to hold a new and different type of tracking activity where everyone would write cards that expressed problems, fears and issues. For 15 minutes everyone anonymously wrote out cards and put them in a pile on a table. Following this initial activity we tried to organize the cards into meaningful groups. Some groups were easy to spot and some were more difficult. Interestingly some of the cards were extremely hard-hitting and potentially hurtful. Of particular note was the card “Hypocritical management”, which seemed strange given the very transparent nature of our XP process and the collective decision making process of the team. While the session was useful at resolving some of the basic problems identified by the cards, it was extremely draining for everyone involved. Along with the CTO, I vowed that we would find a way to do things differently and prevent this kind of buildup from reoccurring. In later discussing this session with a social worker colleague, I was alerted to the problem that anonymous and non-specific complaints are impossible to solve because there is nothing concrete

to test against. I was encouraged to find a forum where specific incidents could be discussed and issues directly addressed.

Retrospectives

A solution to project dissatisfaction is recognized in some of the early XP literature [2], with references made to Norm Kerth's experiences with both large and small technical teams in his retrospective handbook [3]. I first learned about retrospectives at the XP2001 international conference, and after discussing the idea with our CTO, we decided to try one as a follow up to our dissatisfaction tracking session. We started out informally using a "Defining Success Exercise" [3] and discussed how many projects never actually deliver something (by this point we had already launched Sidewize, our first commercial product). This discussion went well, and naturally progressed into discussing the measures described in "How healthy is your organization"[3]. In reading out the characteristics of both functional and dysfunctional teams, we decided on the spur of the moment to create a chart on a whiteboard and measure ourselves on a continuum between the proposed dysfunctional and functional measurements. Our table was subsequently transcribed onto flipchart paper (and slightly abbreviate to fit) and looked like Table 1, with the middle column showing the values we recorded over 3 iterations (marked with a # character).

Table 1. Measurements of team happiness on the axis of dysfunctional to functional behavior.

Dysfunctional Indicator	Team Measure (Iterations 21-23)	Functional Indicator
Guarded Language (secrets)	-5 0 . . # . 5 -5 0 . . ## 5 -5 0 . . # 5	Honest Communication
Distrust of other groups	-5 ## . . . 5 -5 0 ## . . 5 -5 0 . # . . 5	Alliance and cooperation
Well defined boundaries (loss of discussion)	-5 # 5 -5 0 ## . . 5 -5 0 . ## . 5	Boundaries mutually discussed
Blame + Lack of respect	-5 # 5 -5 0 ### . 5 -5 0 . . # . 5	Appreciation + use difference between groups
Skepticism of others ideas	-5 0 . . # 5 -5 0 . . # 5 -5 0 . . # 5	Group refinement of others ideas
Stick (pressure to produce)	-5 0 # . . . 5 -5 0 ## . . 5 -5 0 . # . . 5	Carrot (encouragement to improve)
Living in the past	-5 0 . # . . 5 -5 0 . ## . 5 -5 0 . . # . 5	Creating new solutions

Internal Competition (I look good)	-5 0 . # . . 5 -5 0 . # . . 5 -5 0 . # . . 5	External Success (we look good)
Confrontation	-5 0 . # . . 5 -5 0 . # . . 5 -5 0 . # . . 5	Constructive
No power to change	-5 0 . # . . 5 -5 0 . ## . 5 -5 0 . ## 5	Empowered
Debate to win debate	-5 0 . # . . 5 -5 0 . ## . 5 -5 0 . . . # 5	Consensus (+support, +good enough)
Decision distrust	-5 . . . ## . . . 5 -5 0 . ## . 5 -5 0 . ## 5	Decision respect (trust of skills)
Pressure to conform to standard	-5 0 5 -5 0 5 -5 0 . . # . 5	Flexibility available for new situations

We found this exercise was challenging and very revealing, but after repeating it several times in subsequent retrospectives we found that it became less meaningful (as we achieved high functionality in most measures), and so we opted to stop using it.

Once we had completed this health chart, we then moved onto a “Time Line Mining”[3] activity and asked the questions:

- ?? What we did well
- ?? What we can do better
- ?? What we have learned
- ?? What puzzles us

These questions have formed the backbone for all of our subsequent retrospectives and have not outgrown their usefulness. Over time these titles have been truncated and slightly modified to Well, Not so Well, Puzzles, and Actions. We try to cover the positive aspects first and then move onto the items that we can do better. Often there are puzzles, and it is these, as well as actions, that drive what we will focus on in our next iteration.

Due to the short iterations involved in an XP project, we never felt the need to take the time to formally construct a time line of project events. In retrospect it may have been useful to try this the first time we attempted a retrospective, however in subsequent sessions the time gap is so short that you only need to quickly review the story cards from the earlier part of the iteration (this is a still useful activity to act as a quick refresher).

During our initial attempts at discussing the “mining” questions, we encountered the problem that everyone tried to talk at the same time. Norm described the use of a “coffee mug” as a speaker token, however all our mugs were full at the time. After a quick search I came across our shelf of holiday souvenirs and selected a small stuffed St. Bernard (slightly ironic). This dog was used for many subsequent retrospectives and had the obvious characteristic that you can safely throw it across the room to allow someone else to speak. However, an unusual side effect that we often observed

was that when someone was quite emotional about a subject and began talking, they would invariably begin to unconsciously stroke the toy until they calmed down. Eventually they would realize they were doing this and then would be quite keen to give up the token to someone else.

Of course when someone has the token is discussing new issues, they don't refer back to the previous retrospectives' results until all of the mining questions have been completed for the current iteration. This prevents us from skewing our observations, and makes it interesting to compare results with the previous retrospective to see how things have changed, or how previously important actions have downgraded themselves.

Finally, while it is good to identify actions, we have also noticed that with back-to-back iterations it was often hard to get started on any of them. For this reason we successfully tried modifying our development lifecycle to 3, 1 week iterations followed by a work queue week [7]. In this 4th week we hold our retrospective on a Monday and then have time to address some of the retrospective actions as well as working on stories in a work queue fashion (one of which is gathering new stories for the next iteration).

The Thinking Environment

In our retrospectives, we have also tried looking for alternative approaches, one of which is using a Thinking Environment [4]. While our typical retrospectives were very similar to those described in holding a meeting using a Thinking Environment [4], there were some subtleties that we hadn't quite mastered. Obviously everyone should be given a turn to speak during a retrospective, however this can be challenging as some people are naturally more quiet and don't find an opportunity to speak up. To combat this we adopted the suggested strategy of going round a circle and asking each person to give an item that had gone well in the last iteration. We endeavored not to interrupt each other, and allowed each person to say pass if they had nothing to say. We repeated this until most of the circle had said pass, and then we moved on to things that hadn't gone so well etc.

So far we haven't explored the use of incisive questions to remove limiting assumptions, but this is definitely an area that should be explored. However, we have tried finishing off a meeting with each person offering an item of appreciation to the person on his or her left. This exercise proved to be one of the most difficult we have ever attempted in our retrospectives. It is not easy to look someone in the eye and say something meaningful to them. Equally it is also quite uncomfortable to sit and listen to someone giving you praise. As members of an XP team work particularly closely with each other, it is important to practice these skills. We certainly noticed that following this episode people were much more willing to compliment each other on a daily basis although no one has suggested that we repeat this exercise in a subsequent retrospective. I think this is a shame, but I'm sure the idea will resurface.

The Social Work Experiment

While we have been extremely pleased with our retrospectives, I was still noticing that there was an element of tension that never quite came out during them. Often people would mention things informally, but never together. For a long time I have thought that intense, high-energy teams should have someone around who is an expert in dealing with social situations. After researching the possibilities, and arranging a proposal, I was given the go ahead for a special retrospective to be run by a qualified social worker, which we referred to as a team development day. Both the CTO and I were both slightly nervous about the potential outcome of the day, but felt that it was time to measure our team in a different way to see if it made a difference.

The Introduction

As I wanted to be involved in the experiment I had no knowledge of how the team-building day would run, other than I already knew that our facilitator, James, was a social worker. Before the day started, James arranged the room slightly differently to our normal retrospective, moving the central table to one side and forming a circle with all of the chairs (to seat 12 people). We then began with James introducing himself as a qualified social worker who had been invited to help us run a team-building day. I was interested to note that all the team members seemed very receptive to the idea of having such a person facilitate the day and I also thought that James looked slightly relieved that everyone appeared to appreciate his presence (typically social workers are the last people anyone wants to see at a meeting).

After James' introduction, we all introduced ourselves and described our role in the company and what we hoped to get out of the day. It was interesting to hear what words people chose to describe them, as it's not something we normally do. For example, I recall one team member who has been working in the team for several years, calling himself a junior developer.

Exercise 1

Following the introductions, we divided into two random groups and began the first exercise. Each person was asked to think of 2 truths about themselves as well as one lie. One by one, each person presented their three items and the opposing team then began discussing the pros and cons of each, trying to select the lie. This exercise proved to be very popular but I wondered if the point was to give an insight into the teams' dynamics and see if there were pre-established rungs of power that would need attention. In our case this didn't appear to be the case, as everyone seemed willing to contribute and the identified lie was based on a democratic selection. I later asked James about this exercise, and he replied that actually it was chosen mainly for its merits as an ice breaking activity. However, it does give an insight into how well people know each other - as the discussions indicate levels of both professional and personal knowledge. Furthermore, it also allows the session leader to quickly learn about all the different participants. In our case, James was surprised at how well we

appeared to know each other. This probably stems from continuous pair programming as well as our habit of often eating lunch together.

Exercise 2

Following the warm-up exercise we were then split into 3 groups with the task of presenting the strengths and weaknesses of the entire team. This exercise felt closest to our normal retrospectives but it worked on a higher level and got us to document skills that contribute to the "what went well", and "what didn't go well" questions. While this was a good exercise, as a team we were well practiced on this activity and so the items that we recorded were pretty much things that we had discussed over the previous year. Once again, James was surprised at how well we worked together and identified particular problems that we were able to vocalize amongst each other.

Exercise 3

After the two exercises, we had a lunch break and the mood of the group seemed very positive. When we returned, we began a third exercise to document our work process and highlight areas of communication difficulty. Again we split into three different groups and began drawing process diagrams. As you would expect from an XP team, the diagrams of the three groups looked very similar, documenting the gathering of stories, the planning game, the selection of stories and the day to day working of standup meetings, tasks and pair programming.

Different groups did have a slightly different emphasis on some items, reflecting some of the different skills in each group. For example, a support person (who pairs with a member of the team each day, to form the "exposed pair") documented some of the day-to-day support activities that take place. Another team had a graphic designer who sometimes pairs with members of the team but often works by herself. In this case she drew a separate diagram of how her processes worked, which linked back into the iteration diagram. This latter example was interesting because it was only when she stood up separately to present, that members in other teams slapped their foreheads because they had forgotten her in their diagrams. James later described to me that he had spotted that this designer was having difficulty including herself in the discussion and so had stepped in to suggest that she document her role on a separate paper. This is definitely an important lesson for dealing with mixed skill teams, as you need to encourage non programmers to include themselves in the process even if it means that they document activities that they think don't map directly to XP.

During the presentations it was also pleasing to notice that people who wouldn't normally stand up and speak directly to the group, had decided to take the floor and present. When we all had a drink later in the week, several other people in the team mentioned that they had been amazed that some of the quieter members had made a point of stepping forward to make very effective presentations. It seemed that having a social worker available as a facilitator made people more comfortable taking personal risks. Additionally, during these presentations and discussions, I noticed that James would discretely change his seat in the room. When I later asked him about

this, he explained that by changing positions he was able to establish a better rapport with the group and get a different perspective on ideas being presented. He also commented that he was impressed with how closely each of our diagrams matched each other, as typically team members have radically different ideas of how their process works. Surprisingly, other team members later described how useful this particular exercise was to them, as they hadn't actually taken the time to think about our process. They knew it by rote (having joined an XP team and being immersed in that culture) but hadn't taken the time to think about the implications of each of the steps.

Triangle of Power

During the final presentation, a discussion arose about "seeking permission" to perform certain activities. This kind of discussion had surfaced earlier in the day but the facilitator had steered the group away from it. This time however, he let the group explore this topic and it emerged that three senior team members had differing views and wanted to debate them. At this point the facilitator let the conversation continue for a few minutes and then pointed out to the rest of the group that there was an interesting triangle of power in the team that should be recognized. James further went on to comment that it was perfectly normal to have a disagreement in the team and that other members should not be afraid of this kind of situation. He then went around the group and asked everyone what their opinion on the topic was. This simple technique made a big difference to the mood of the group, and catered both to the fears of the observing team members and the frustrations of the senior members who wanted to try talking about specific trickier issues. In a later conversation with James, he explained that by identifying this triangle of opinion it made it clearer that those involved in the triangle were not necessarily arguing but debating a difficult problem. Furthermore, as a facilitator he mentioned that it's important to notice when other people in the group are beginning to disengage and to bring them back into the conversation by asking everyone for their opinion. This is similar to the "Dealing With Conflict Procedure outlined by Norm Kerth [3].

Revelation

One of the most revealing items from this experiment was that James felt that as a team we were much better equipped to deal with team problems than most social work teams (who ironically have specific training for social situations). He was also impressed with the openness and enthusiasm that we have managed to maintain on the team, even after more than three years of working together.

Following this experiment, we have noticed that many more members of the team are willing to get involved in the more difficult decisions that have to be made in the evolution of the Connextra products. There is definitely a sense that some of the problems that we feared we had, have been accepted as normal and everyone is more willing to discuss the more difficult issues that remain unresolved.

Summary

While this paper outlines the history of a well-established XP team, it also describes some techniques for maintaining a stable and long lasting working relationship. From using collective ownership as a way of hiring the right staff, right through to monitoring iteration velocity for indications of unrest, you need to be ready to apply these techniques when the need arises. Just as when we resolved to never let things get so bad that anyone could even be considered a “hypocritical leader”, so too must you be willing to adopt practices like retrospectives, and group appreciation. Furthermore, it should not be a scary notion to allow experts from other disciplines like Social Work, examine your process for both weaknesses and strengths. It may turn out that you will be pleasantly surprised at how well the simple XP practices can all work together to provide an enduring team. In the future, we hope to report on using the social work experiment with other teams to compare results with those described here.

References

1. Kent Beck. *Extreme programming explained: embrace change*. Addison-Wesley, 1999.
2. C Collins, R Miller. *Adaptation XP Style*, Proc XP 2001
3. Norm Kerth. *Project Retrospectives, a handbook for team reviews*. Dorset House, 2001
4. Nancy Kline. *Time To Think*. Ward Lock, 1999
5. Giancarlo Succi, editor. *Extreme Programming Perspectives*. Addison Wesley, Aug 2002
6. Giancarlo Succi, M. Marchesi, editors. *Extreme Programming Examined*, Addison Wesley, 2001
7. John Vlissides, James O. Coplien, Norman L. Kerth, editors. *Pattern Languages of Program Design 2*. Addison Wesley. 1996

Appendix: Acknowledgements

The author would like to thank: the team at Connextra for being willing to try new approaches. John Nolan (Connextra CTO), who spent considerable time “after hours” in the Highgate, listening and contributing to most of the ideas presented in this paper. Alix Walton and James Blewett who were the social workers that enthusiastically agreed to try the cross discipline experiment. Marston, Thompson & Evershed who brewed the “Pedigree” that enabled many of the conversations to happen in the first place.