Adherence Survey – eXtreme Programming (XP)

Team/Project: _____ Date: __/__/__

A) Years of formal education in IT or Computer Science
(To answer this question, consider the normal duration of the course, regardless of how long it took you to achieve the degree, i.e. if you have completed your undergraduate degree and your first year of graduation your answer should be "5 years" (4 years of undergraduation + 1 year of graduation, even if it took you 6 years to achieve your undergraduate degree)

( ) 0 ( ) 1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( ) 6 or more

B) Years of work experience in the IT industry

( ) 0 ( ) 1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( ) 6 ( ) 7 ( ) 8 ( ) 9 ( ) 10 or more

C) Number of Object-Oriented Programming and related courses
(e.g.: OOP, Design Patterns, Distributed Object Systems, etc.)

( ) 0 ( ) 1 ( ) 2 ( ) 3 ( ) 4 or more

D) Rate your answers to the questions from 1-10 (10 is the highest) considering the current level of your team (Current) and what you consider to be the appropriate level (Desired) for each eXtreme Programming (XP) practice:

Practice	Comments
Pair Programming Current: _____ Desired: _____	*Two people work together at one computer. They trade turns typing or reviewing and thinking about the big picture.* 10 We wouldn't want to write any critical code without pairs taking turns thinking or typing. We rotate pairs. 8 We often work in pairs. It gives us some ergonomic relief. 6 We often have whiteboard chalk talks, chat messages, or office visits. Some people pair program at the keyboard, but some prefer not to try it. 4 We try to pair but can't due to flextime and meetings. Some people are just too slow or fast for me to have patience to sit with. Our furniture makes it hard anyway. 2 I find it distracting when people interrupt me. My office mate asks me not to have so many visitors. 0 I wear earphones so people won't disturb me. Actually, I prefer to work at home with the phone off the hook and my chat program set to do not disturb.
Small Releases Current: _____ Desired: _____	*We deliver more frequent, smaller iterations to customers.* 10 Every week or two the customer can take the system as a working unit. 8 We have one month iterations. They can pick new function for the next iteration at that time. 6 Every few months we have another iteration for the customer. 4 We ship beta drivers and fix packs about 4 times a year, with bigger releases in 8-12 month cycles. 0 We have a grand vision. Next year's release 1.0 will tide you over until the real function comes out in release 2.0 in the 18+ month timeframe.
Continuous Integration Current: _____ Desired: _____	*When working on a big set of code I sync up or checkin as follows:* 10 Several times per day. 8 Once per day. 6 Several a week. 4 Once a week. 2 A few weeks can go buy. I only check it in when it's ready. 0 I usually have problems because a lot of changes have occurred between the time I check it out and the time I try to sync up. I have to resync several times because stuff is back leveled or left out. The build seems to break often.

Test-Driven Development Current: _____ Desired: _____	*Do we have test cases and automated drivers for each product class?* 10 The automated tests are the design. The customer runs acceptance tests. 8 After we design and prototype the code we write some testcases. 6 We are careful to unit test our code after it's done before it goes to the test team. 4 We've heard of tools like JUnit, but haven't really tried it. 2 Our formal system test phase at the end of our cycle takes much longer than planned because there are so many bugs and fixes going in. 0 We don't really have any formal testing. Customers often let us know if there are any problems though.
Planning Game Current: _____ Desired: _____	*We move items in and out of the plan based on updated customer needs while keeping the dates steady.* 10 Before every short iteration the customer chooses the most valuable features based on the developer's sizings. Each morning we review the outstanding user stories in a 5 minute stand up meeting and volunteers pair up to implement them. Since we know change happens, we have a competitive advantage because we've optimized our process to accept and exploit change. 8 We trade function in and out of our milestone drivers from time to time after customers change their priorities and development is completed ahead of time or falls behind for some items. It's the customer's product after all. They should get what they want. Change happens. 6 Plans should not change. We meet our dates, even when planned a year in advance We create a lot of 'artifacts' like design specs. We try to keep them up to date. 4 We are careful to follow the 'waterfall' process. We don't start coding until the designs are complete and reviewed. We don't start test until all the code is delivered. Sometimes we've changed or missed our dates because the customer changed a requirement. 0 We lose customers because we tell them they'll have to wait for the next release. After all, we're busy finishing what they asked for last year.
On-site Customer Current: _____ Desired: _____	*We have access to our customer and get feedback from her.* 10 We don't consider a line item done until the customer has run their acceptance test. 8 We frequently interact with customers to show them prototypes to see how they want it changed. 6 We get requirements from customers. 4 Requirements come from somewhere, but I don't think their customers. 2 We ship functions but are never sure if we made what they wanted. They've probably changed their mind by now anyway. 0 We know what's right. They'll use our stuff weather they like it or not. It'll be good for them.
Refactoring Current: _____ Desired: _____	*We rewrite or redesign code that smells bad or will help position us for new requirements.* 10 We often use it as a tool to steer or flex the design to meet changing requirements. We also do it to make it more streamlined and easy to change in the future. 5 We've done some cleanup from time to time. 0 We have old baggage code that has had a lot of ungraceful changes. We're afraid to touch it. We turn down requirements because the code is 'works as designed' and can't be changed.
Simple Design Current: _____ Desired: _____	*We keep the design simple now so we can change it as needed in the future.* 10 YAGNI (You ain't gonna need it). We often refactor needed changes in later. 8 It's clean - it does just what's needed in the simplest way. 6 Our design is mostly straight forward, with a few lumpy spots. 4 We've proudly engineered a full feature, full function product that does everything people will need. 2 We have big chunks of code that are not finished or end up being thrown away. 0 I thought I'd write this framework incase we need it in the future. You never know.
Coding Standards Current: _____ Desired: _____	*Do you have and follow enough standards so you can read and change each other's code? Have 'em? Detail? not too much, not too little?* 10 We have standards, follow them, train new people in them. By they way, they are industry standards as well. 8 We have most standards written down, and most people usually follow them. 6 We do the same stuff for some things, but some things are treated differently. Our braces are in the same place, but our error handling is often different. 4 We have many standards and each follow a different one. 2 We don't have standards. 0 How dare you tell me what to do?

Collective Code Ownership Current: _____ Desired: _____	*People can change each other's code. We don't have to wait for the specialist.* 10 We regularly change code in any area. You can't tell because our code looks the same. 8 We regularly change code in any area. 6 We've changed each other's code, but usually assign stuff in specialty areas. 4 We can fix it if we have to. 2 We'll have to wait for them to get back from vacation. 0 I lock all my files and keep 'em locked.
Metaphor or System of Names Current: _____ Desired: _____	*So you have naming conventions for your objects?* 10 I can tell what everyone's code does from the name, without looking at the comments We often think of the same name for things. When discussing design we phrase the discussion on a common metaphor for the system. 8 I like the names in the system. They make sense to me. 6 I can follow the names, with a little help from the comments. 4 The names are misleading, You really have to read the code. I'm not sure what to name new classes. 2 You have to understand the history, the names are one thing but the methods to different stuff because evolved differently. 0 Why did they call it that? I can't read their abbreviations.
Sustainable Pace Current: _____ Desired: _____	*Do people work at a rate that's effective for them over the long run?* 10 We work a steady comfortable pace. We sprint only when needed, which isn't often. 8 Sometimes I'm too busy, sometimes too bored. 6 Whenever there is a deadline we go into death march mode. 4 We've been ordering dinners for several months now. It seems it's always like that. 2 More than once I've had to cancel my vacation or classes. 0 I don't have time to fill out this survey.
Lessons Learned Current: _____ Desired: _____	*Do you stop periodically to consider ways to improve?* 10 People often come up with new ideas, and they are implemented. We share the new techniques with other groups. 8 We think of what went wrong, what went right, and make changes for our team. 6 We come up with ideas but they seem to die on the vine and never get into the culture. 4 We repeat the same mistakes. 0 We never have 'em. I have gripes and they never seem to get addressed. Obviously no one cares.
Tracking Current: _____ Desired: _____	*There are big visible charts spread over the wall that helps us understand the project pace.* 10 We have several charts that are updated daily and we remove the ones that are not being used anymore. The charts help us understand and improve our process. 8 We have some interesting charts on the wall that are updated weekly. 6 The information in the wall is updated at the end of each release. 4 The charts are outdated and no one cares about them anymore. We have to work to finish on schedule. 2 I do not know why we have those charts on the wall. They do not seem to be related to my work. I think no one would notice if they were removed. 0 We do not have any charts on the wall. We think it is better to store important information on documents and files in our central repository.
Overall XP Score Current: _____ Desired: _____	*To what extent to you feel you implement XP practices?* 10 I coach XP, have written books, or presented at conferences. 8 We're a good example of an XP team. 6 We often fold XP practices into our daily routine. We've read about XP, but since lives or huge amounts of money depend on our software and we have a very large team, we use formal methods with lots of documentation and reviews. 4 I don't know what XP is, but the scoring principles on this survey sound intriguing. Maybe we can try them sometime. 2 I'm not sure what XP is, but it sounds bad. 0 I don't believe software development should be tied down by a 'process' – or – We've always done business the same way. I see no need to change now.

Final Observations / Suggestions:

_____

_____