

# Are we doing XP?

---

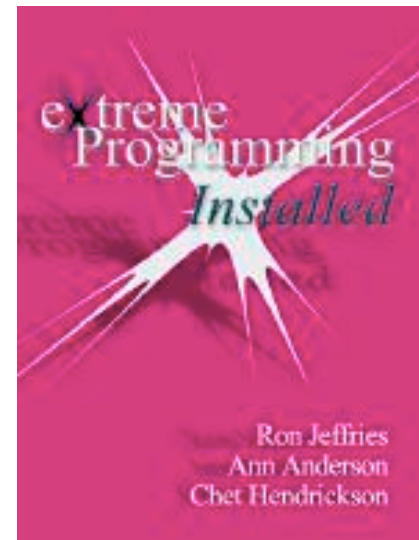
How can we know, and does it matter?

**Ron Jeffries**

**[www.XProgramming.com](http://www.XProgramming.com)**

**[www.ObjectMentor.com](http://www.ObjectMentor.com)**

**[RonJeffries@acm.org](mailto:RonJeffries@acm.org)**



# Does it matter?



- † **Well, we don't have a customer ...  
So we can't have acceptance tests.**
- † **We're too busy fixing bugs ...  
To do much planning  
Or to write many unit tests**
- † **But we don't write any documents.  
So we're doing XP!**

# How can we tell?



† Rights and Responsibilities

† Practices

† Results

# Customer and Manager Rights

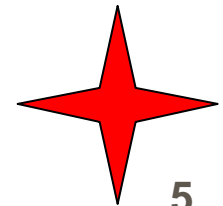


- † You have the right to an overall plan, to know what can be accomplished, when, and at what cost.
- † You have the right to get the most possible value out of each programming week.
- † You have the right to see progress in a running system, proven to work by passing repeatable tests that you specify.
- † You have the right to change your mind, to substitute functionality and to change priorities, without paying exorbitant costs..
- † You have the right to be informed of schedule changes, in time to choose how to reduce scope to restore the original date. You can even cancel at any time and be left with a useful working system reflecting investment to date.

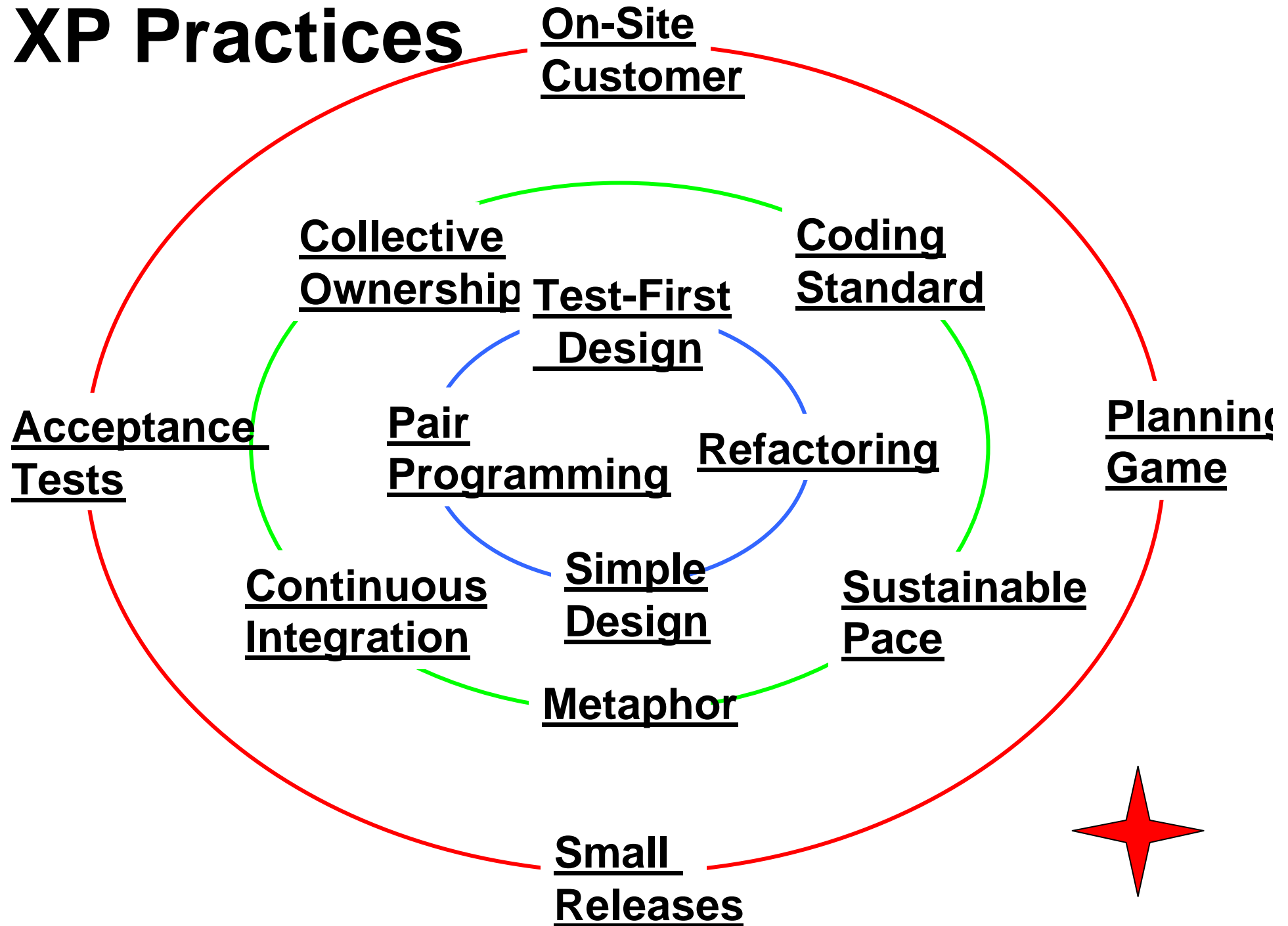
# Programmer Rights



- † You have the right to know what is needed, with clear declarations of priority.
- † You have the right to produce quality work at all times.
- † You have the right to ask for and receive help from peers, superiors, and customers.
- † You have the right to make and update your own estimates.
- † You have the right to accept your responsibilities instead of having them assigned to you.



# XP Practices



# Planning Game



- † User Stories
- † Release Plan
- † Iteration Plan
- † Reporting

# Planning Game

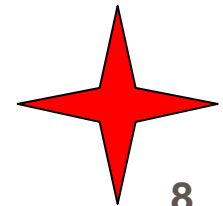


## † Business decides ...

- † Scope
- † Priority
- † Release Composition
- † Release Dates

## † Technical decides ...

- † Estimates
- † Consequences
- † Development Process

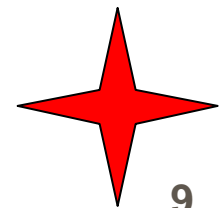




# Small Releases



- † **Deliver real business value**
- † **On a very short cycle**
  - † **Business value sooner**
  - † **Rapid feedback**
  - † **Sense of accomplishment**
  - † **Reduced risk**
  - † **Customer confidence**
  - † **Adjustments to requirements**



# Metaphor



✚ Does it matter? Yes.

✚ Can I help you? Well ...



# Simple Design



- † The right design for the software is one that
  - † Runs all the tests
  - † Has no duplicated logic.
  - † States every intention important to the programmers.
  - † Has the fewest possible classes and methods



# Test-First Design



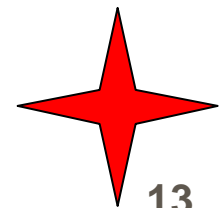
- † Unit tests for all classes
- † Written immediately before the code
- † Upgraded on every defect
- † Run before any release
- † Always score 100%



# Acceptance Tests



- † **Very Important**
- † **Owned by customer**
- † **Test every story**
- † **Catch regressions**



# Refactoring



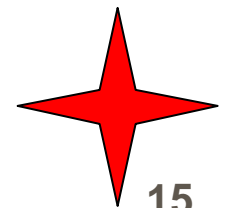
- † If you are doing more and more refactoring:
- † ... you aren't doing enough refactoring!
- † Runs All the Tests
- † Contains No Duplication
- † Expresses Every Idea



# Pair Programming



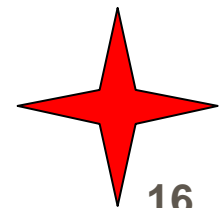
- † All production code is written with two people sitting side by side at the computer.



# Collective Ownership



- † Anybody who sees an opportunity to add value to any portion of the code can do so at any time.
- † Everyone is responsible for the whole of the system.





# Continuous Integration



† A clean system can be shipped at any time

† Build the system Multiple Times per Day!!



# Sustainable Pace



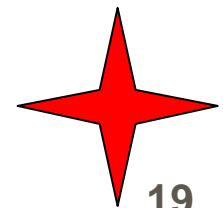
- † Death March?
- † High Pressure?
- † Long Hours?



# On-Site Customer



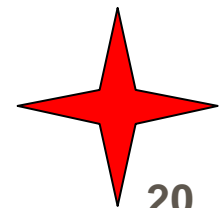
- ✚ A real customer must sit with the team, available to answer questions, resolve disputes, and set small-scale priorities.



# Coding Standard



- † **Make it impossible to tell who wrote what.**
- † **Emphasize communication.**
- † **Adopted voluntarily by entire team.**



# Have you ever worked on a project ...



† ... where before a programmer added even one line of code he wrote a test that showed that the line was needed ... and that it worked when added?

# Have you ever worked on a project ...



† ... where every time anyone found a problem, a test was written to show that the problem existed ... and then that it was fixed?

# Have you ever worked on a project ...



† ... where every time the customer came up with a requirement, the testers or programmers helped her come up with a repeatable test of whether the requirement was correctly implemented?

# Have you ever worked on a project ...



† ... where every time a requirement wasn't clear, someone who understood the requirement was right there, ready to explain it?



# Have you ever worked on a project ...



† ... where every time anyone found a problem, the team met to figure out what went wrong and how to avoid such things in the future?

# Have you ever worked on a project ...



† ... where every class and every method had automated tests running at 100% correctness all the time?

# Have you ever worked on a project ...



† ... where the software was integrated and fully tested multiple times every day?

# Have you ever worked on a project ...



† ... where every single line of code was reviewed by at least one person in addition to the programmer who wrote it?

# Have you ever worked on a project ...



† ... where every bit of redundancy was removed from the code as soon as it was detected?

# Have you ever worked on a project ...



† ... where every programmer spent every moment working only on things that were the most important to the customer?

# Have you ever worked on a project ...



† ... where the design of the system was consciously and carefully improved every day?

# Have you ever worked on a project ...



† ... where every day everyone did their best to let everyone hear what was learned yesterday?



# Have you ever worked on a project ...



† ... where the software was fully ready for release every week, or even every day?

# Have you ever worked on a project ...



† ... where everyone worked hard, but kept themselves rested and healthy for the long haul?

# Have you ever worked on a project ...



† ... where everyone on the team – even the customers and managers – knew how the system worked and could explain it?

# Have you ever worked on a project ...



† ... where the design was always just right for what the system did – neither too simple nor too complex?

# Have you ever worked on a project ...



† ... where every programmer was able, empowered and encouraged to improve any part of the system at any time?

# Have you ever worked on a project ...



† ... where the code was of such uniformly good style and quality that it all looked as if you yourself, god that you are, had hand-crafted it?

# Have you ever worked on a project like that?



† I have.

† Does doing XP matter?

† Oh yes, it matters.

† Are you doing XP?

# Are we doing XP?

---

**How can we know, and does it matter?**

**Ron Jeffries**

**[www.XProgramming.com](http://www.XProgramming.com)**

**[www.ObjectMentor.com](http://www.ObjectMentor.com)**

**[RonJeffries@acm.org](mailto:RonJeffries@acm.org)**

