Go free RANGE

Pair Programming

Overview

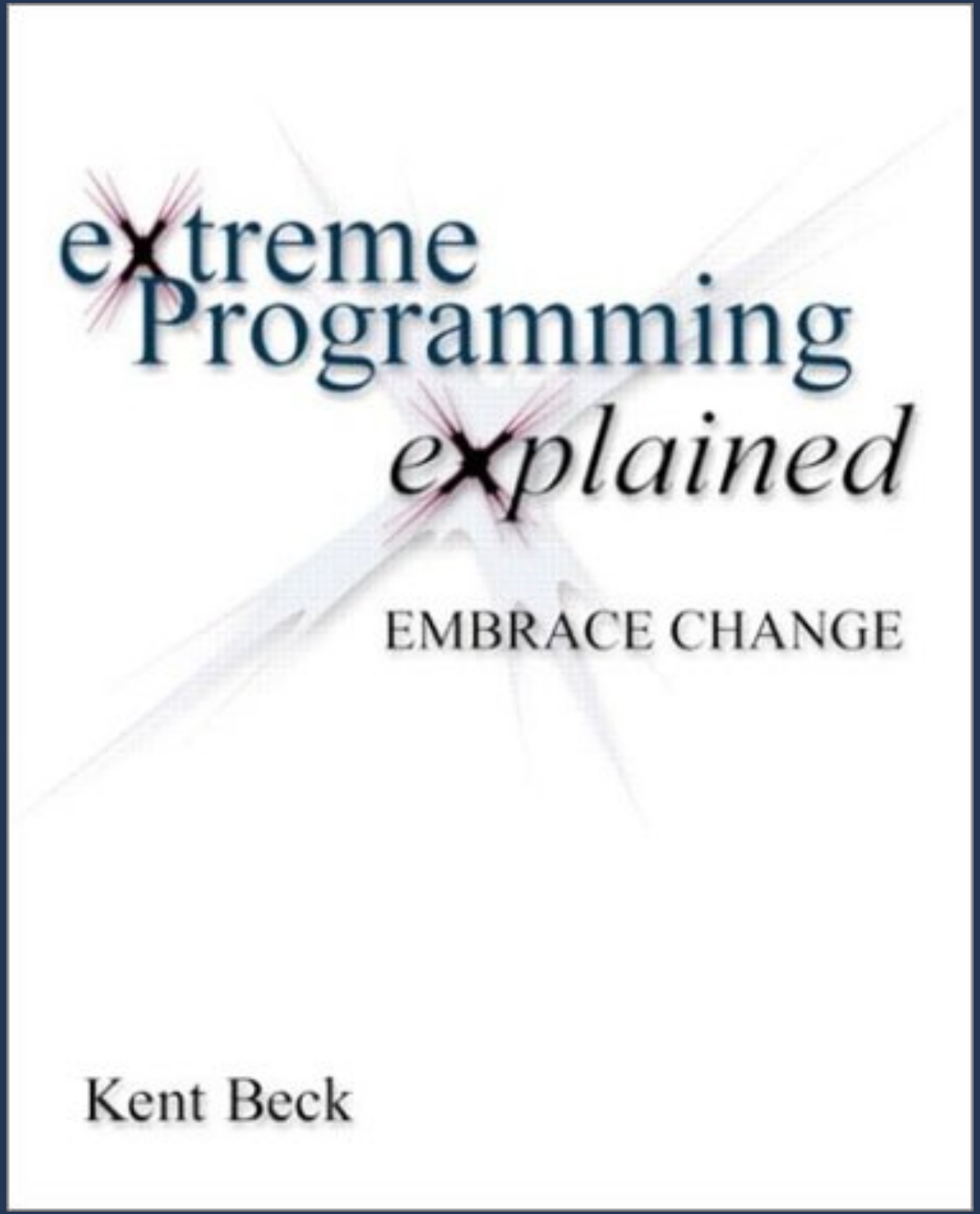
- What is pair programming?
- Why we think pair programming is a Good Thing™
- Why we sometimes find it hard to **start** pairing
- Why we sometimes find the **process** of pairing difficult
- Q&A

Disclaimer

- Not a prescription
- Not a criticism
- Just our opinions

What is Pair Programming?

XP: The "White Book"



extreme Programming explained

EMBRACE CHANGE

Kent Beck

All production code is written with two people looking at one machine, with one keyboard and one mouse.

There are two roles in each pair.

One partner, the one with the keyboard and the mouse, is thinking about the best way to implement this method right here.

The other partner is thinking more strategically.

Pairing is dynamic.

A dialog between two people trying to simultaneously program.

A conversation at many levels assisted by & focused on a computer.

A subtle skill - you can spend the rest of your life getting good at.

What Pair Programming is Not

One person programming while another watches.

A one-way tutoring session.

One person judging another.

Something only less experienced people do until they're up to speed.

Why we think pairing is a Good Thing™

XP Practices (1-6)

- The Planning Game
- Small Releases
- Metaphor
- Simple Design
- Testing
- Refactoring

XP Practices (7-12)

- Pair Programming
- Collective Ownership
- Continuous Integration
- 40-hour Week
- On-site Customer
- Coding Standards

XP Practices Supporting Pairing

You can't possibly write all the production code in pairs. It will be too slow. What if two people don't get along? Unless:

- Coding Standards
- 40-hour Week
- Testing
- Metaphor
- Simple Design

Then perhaps you could write all production code in pairs.

Besides, if people program solo they are more likely to ***make mistakes****, more likely to* ***overdesign****, and more likely* ***drop the other practices****, particularly*

XP Practices Supported By Pairing

- **Simple Design**
- Testing
- Refactoring
- Collective Ownership
- Continuous Integration

- Simple Design
- **Testing**
- Refactoring
- Collective Ownership
- Continuous Integration

- Simple Design
- Testing
- **Refactoring**
- Collective Ownership
- Continuous Integration

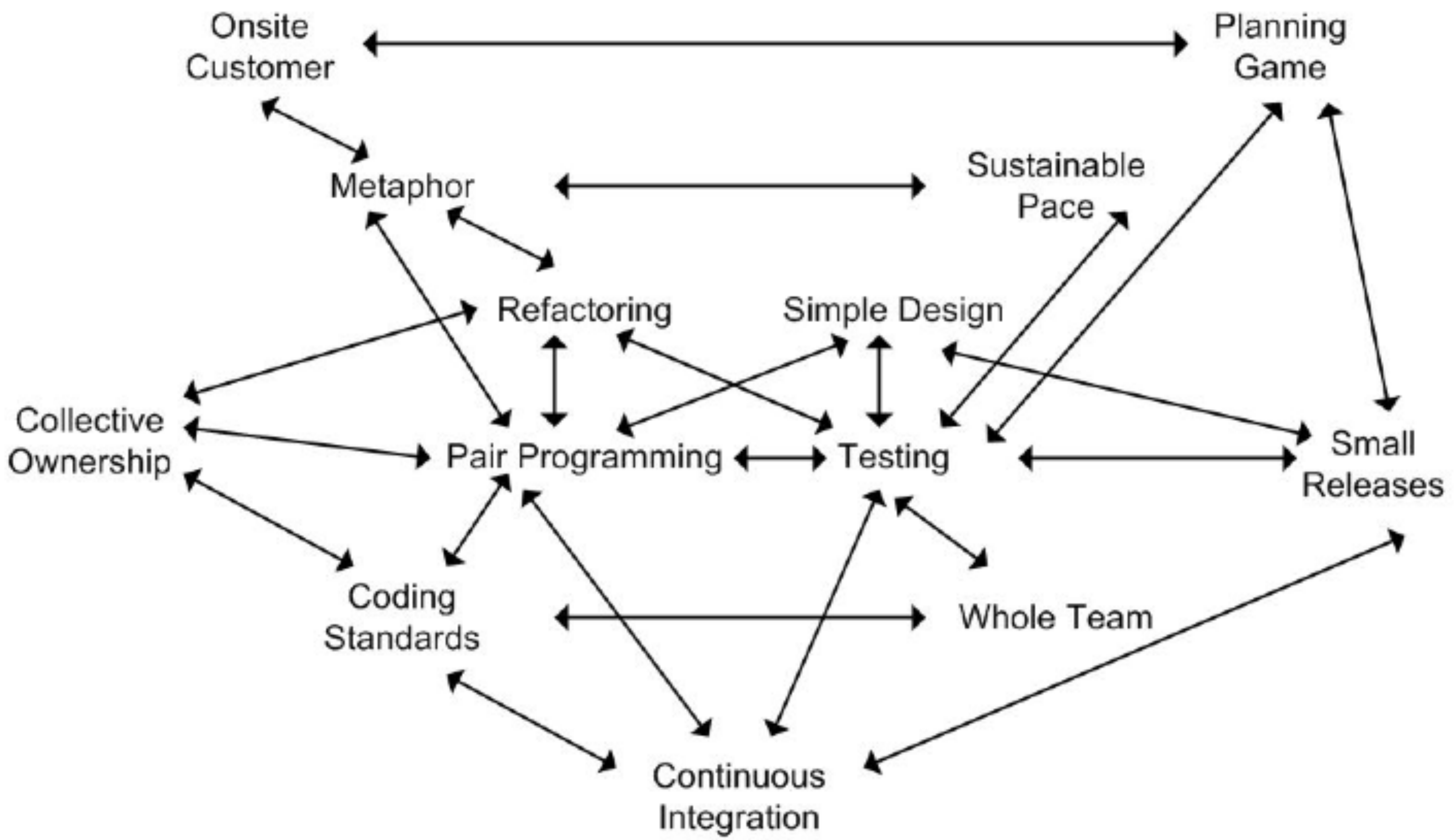
- Simple Design
- Testing
- Refactoring
- **Collective Ownership**
- Continuous Integration

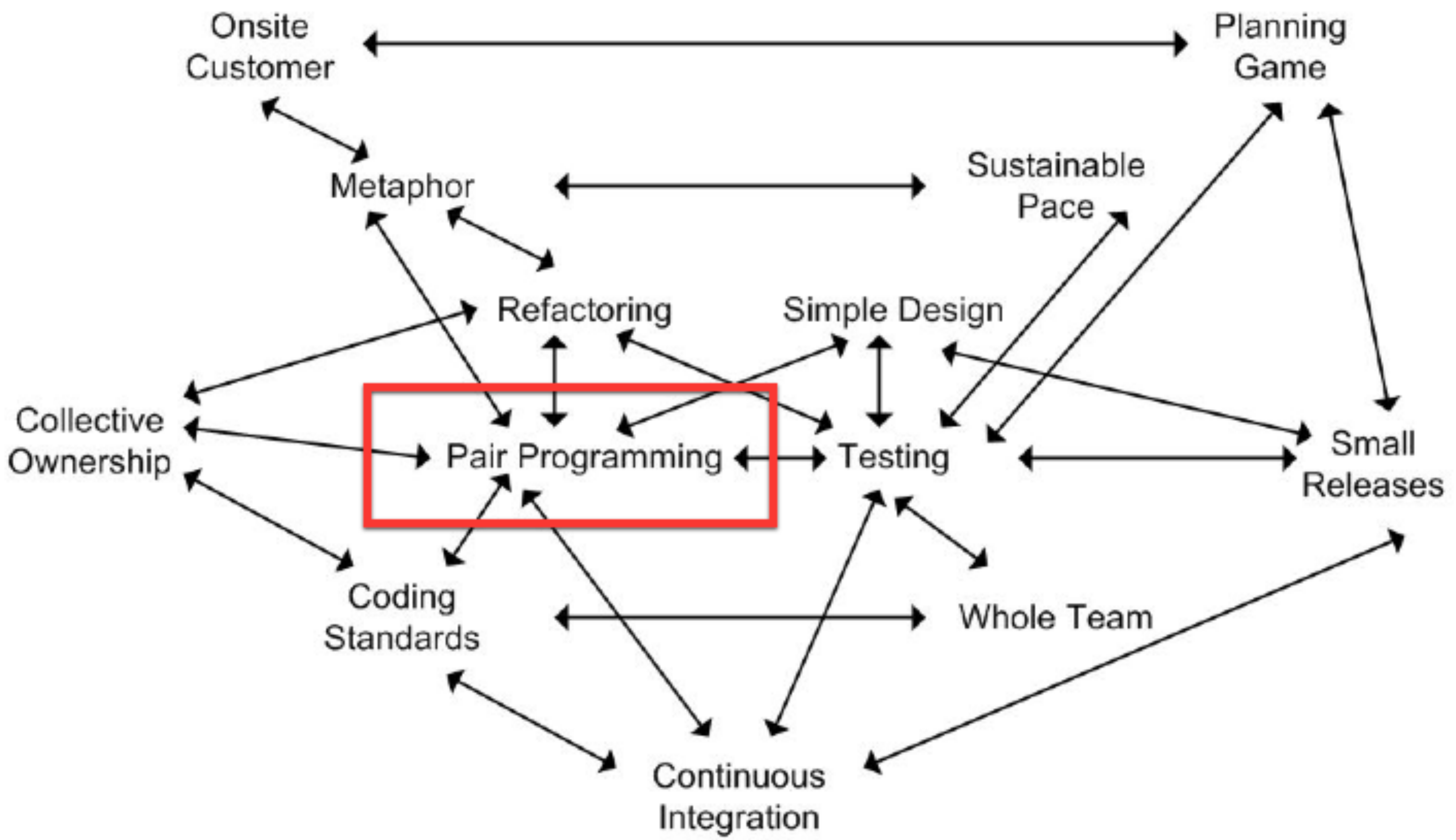
- Simple Design
- Testing
- Refactoring
- Collective Ownership
- **Continuous Integration**

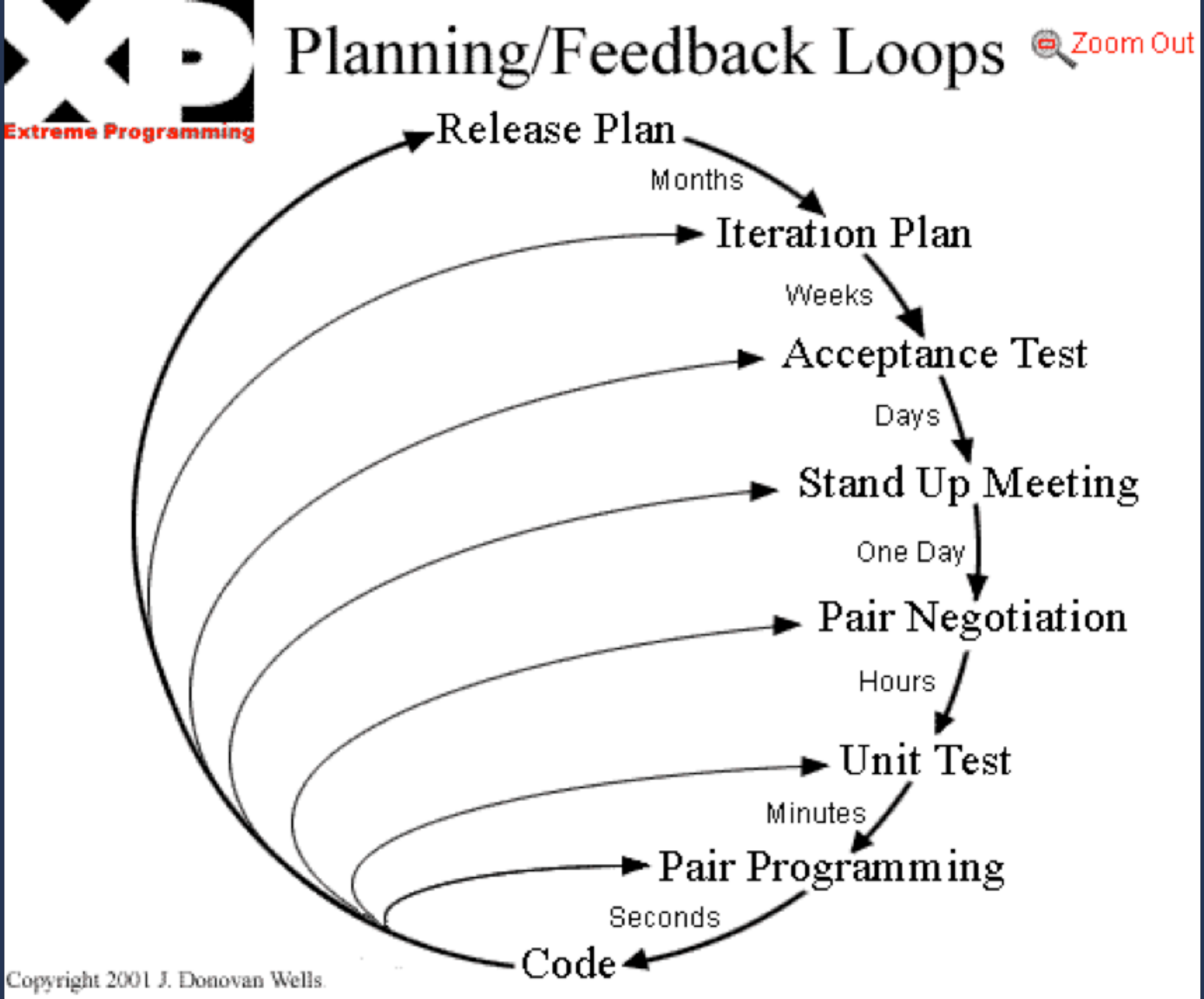
Mutually Supporting Practices

The individual pieces are simple. The richness comes from the interactions of the parts.

– Kent Beck


Onsite Customer
Planning Game
Metaphor
Sustainable Pace
Refactoring
Simple Design
Collective Ownership
Pair Programming
Testing
Small Releases
Coding Standards
Whole Team
Continuous Integration


Onsite Customer
Planning Game
Metaphor
Sustainable Pace
Refactoring
Simple Design
Collective Ownership
Pair Programming
Testing
Small Releases
Coding Standards
Whole Team
Continuous Integration



Planning/Feedback Loops

Why we sometimes find it hard to start pairing

I feel intimidated to work with someone more experienced

- *Nobody* knows what they're doing
- The impostor syndrome
  - "I'm just making stuff up as I go along"
  - "I'm going to be found out"
- The Dunning-Kruger effect
  - "There's so much that I don't know"

I don't want to disturb someone to ask them to pair

- Agree explicit rules about interruption for pairing
- Fixed pair rotation times e.g. every day/half-day
- No need for same pair to work on same story from beginning to end

No other developer available to pair e.g. stuck in meetings

- Core pairing hours
- Doing more pairing ought to reduce the need for as many meetings
- Shorter more efficient meetings / make attendance voluntary
- Spike on something / refactor something

I found my last pairing session hard work

- Be patient - pairing is a skill which takes time to learn
- Decide to try a new tactic
- Pair with someone different
- Encourage your colleagues to pair more often

Why we sometimes find the process of pairing difficult

Exhaustion / fatigue

- Take regular breaks

Difficult to articulate thought processes

- Sketching on a whiteboard / piece of paper
- Pseudo code sketching

No time for personal admin

- Don't try to pair all day

Some types of task are hard to pair on e.g. writing commit notes/documentation

- Allow a single person to write and have the pair review it at the end

One person hogs/avoids the keyboard

- Pairing ping-pong

One person is distracted by email/IM notifications

- Switch off all distractions

Differences in development environment e.g. editor, shell, OS

- Agree on a common development environment

Physical environment not conducive to pairing

- Be prepared to move furniture (e.g. drawers)
- Make your pair feel comfortable
- Consider using two mirrored displays

Q&A

References

- http://www.extremeprogramming.org/
- http://codon.com/i-have-no-idea-what-im-doing
- http://en.wikipedia.org/wiki/Impostor_syndrome
- http://en.wikipedia.org/wiki/Dunning%E2%80%93Kruger_effect