

# Practical Tips for Pairing with Developers to Test

Toby Sinclair



@TobyTheTester



# What you may learn?

**What** is pairing?

**Why** should you pair?

**Who** should you pair with?

**How** can you go about pairing?

**Where** you might go wrong!

# What is pairing?

*”Two people(or more) approaching a problem where either person, may or may not know the solution, but both appreciate that having another persons skills and perspective, may help to resolve the problem either quicker or better.”*

What do you think?

# Different Kinds of Pairing

- Pair Programming - Typically Two "Developers"
  - *Solving coding problems together*
- Pair Testing - Typically Two "Testers"
  - *Doing testing together*
- 3 Amigos
  - *BA, Developer, Tester having conversations to clarify stories*
- Mob Programming
  - *Whole Team working on Same Problem using the same Workstation*

# Why should you pair?

- **Learn** about the product/software
- Helps move towards a **shared understanding**
- Problems can be solved **quicker** (Twice as fast) [1]
- **Builds relationships** within teams
- It's **fun!**

[1] Beck, K., Extreme Programming Explained: Embrace Change. 2000, Reading, Massachusetts: Addison-

# Who should you pair with?

- Developers
- Testers
- Customers
- Business Analysts
- Operations
- *“Team Members”*

# What stops you pairing?

- Management

- It's not always natural -  
We forget to pair

- Team members not co-located

- Team member Sprint Tasks are not aligned  
(*Development Sprint / Testing Sprint*)

So **how** can we pair?





# Pair Testing

*“Pair testing is different from many other kinds of pair work because testing is an **idea generation activity** rather than a plan implementation activity. Testing is a **heuristic search**. Pairing has the effect of forcing each tester to explain ideas and react to ideas. When one tester must phrase his thoughts to another tester, that simple process of phrasing seems to bring the ideas into **better focus** and naturally triggers more ideas.”*[1]

[1] Exploratory Testing in Pairs – <http://www.kaner.com/pdfs/exptest.pdf>

# Use tools to support pairing



- Development Debug tools e.g. Visual Studio
- Physical Tools e.g. Two Keyboards
- Tests! e.g. Unit Tests
- Examples (BDD)
- Thinking Tools (e.g. Six Thinking Hats)
- Remote Pairing Tools

# Gaining Agreement to pair

- What if my team say no?
- Begin in small increments.
- Demonstrate the benefits
- Build a working relationship
- Use influencing skills
- Agree a suitable time and frequency



# An alternative approach?

<https://www.youtube.com/watch?v=dYBjVTMUQY0>

# What to do with odd socks?

- Not all pairs work.
- It can be emotionally challenging!
  - Personalities
  - Experience Levels
  - Do it my way or the highway!
- How to deal with conflict?
  - ELMO (Enough, Lets Move On)
  - Pair doesn't have to mean two. Bring in an independent person

# Learn from and do Exploratory Testing

- Exploring with someone else can be more fun than on your own!
- Exploratory Testing structure supports pairing
- Use charters
- Note taking easier with more than one person (e.g. Mind Maps)
- Record your sessions ([pairwith.us](http://pairwith.us))



# Influence Testability



- Introduce Testability as you build
- Observe areas that could be made more testable
- Ask for changes and if they are simple, observe the changes there and then
- Shared Understanding of the importance in Testability

# Different Kinds of Testability

- |    |                 |
|----|-----------------|
| 1. | Intrinsic       |
| 2. | Project Related |
| 3. | Value Related   |
| 4. | Subjective      |



To Pair or Not to pair that is  
the question!



# Use Heuristics!

*“A heuristic is a fallible method for solving a problem or making a decision.”*

# When to pair?

These are the “Starting” heuristics i have used to determine when to pair:

- Lone Ranger Heuristic

*“Working alone on a problem desperately trying to find and make sense of information at hand.”*

- Alien Heuristic

*“I don’t know much about this team member, i’d like to work with them to learn more about how they think.”*

# When not to pair?

It's not always a good choice to pair. There are problems which are better solved alone. These are some heuristics i use:

- No need for L Plates

*“You know what needs to be done and are confident of hitting the road alone”*

- In the basement

*“You want to work on a problem with no influence from another person, to avoid the influence their personal bias, may have”*

# Warning!

Heuristics are fallible!

The time you decide you don't need to pair could be  
the time you really should pair!

# Where you might go wrong?

- Shallow Agreements
- Group Think
- Thinking your pairing when you are really just “helping”
- Watch the Master!
- Stop Pairing at the time when you might need it most!

*For example, meeting a Deadline!*

# Putting it all together

<https://www.youtube.com/watch?v=rwQqkX3qZak>

# Questions?

@TobyTheTester

