



Pair Testing

Jonathan Kohl
jonathan@kohl.ca
www.kohl.ca/blog



What is Pair Testing?

- As in pair-programming, two people simultaneously test an application at the same computer
- It can involve:
 - testing an application from a customer's perspective
 - more technical kinds of tests



Differs from Pair Programming

Pair Programming:

- Output is documented code

Pair Testing:

- Output is information
 - Bug Reports
 - Test Strategies
 - Project Status



Why Pair Test?

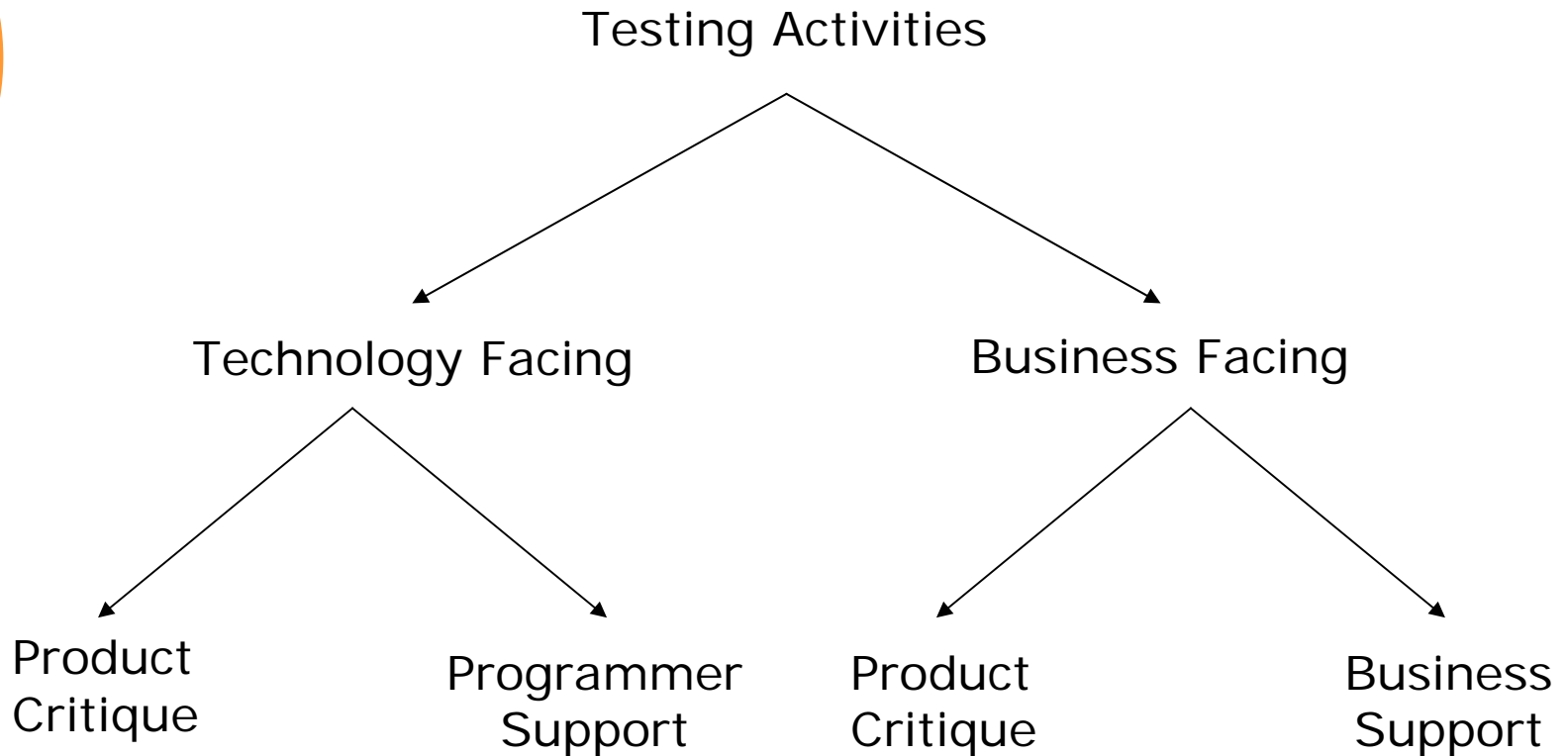
- Software testing is primarily an idea-generating activity which is enhanced by pairing –*Kaner & Bach, Exploratory Testing in Pairs*
- “Test-infected” developers who are automating tests can learn more about testing techniques
- Dedicated testers can learn a tremendous amount about testing from developers



Who can Pair Test?

- Dedicated Testers
- Developers
- Documentation Writers
- Customers
- Analysts
- Any combination of the above

When to Pair Test



cf. Brian Marick's "Agile Testing Directions"



Technology Facing Activities

Programmer Support:

- Testers work on tasks to assist development (such as automating user acceptance tests, developing FIT-driven test cases, exploratory test areas of risk.)
- Testers and developers pair together:
 - during development (to generate test ideas and ensure the code is testable)
 - after development (to ensure the code is testable and that the interfaces can be tested against)



Technology Facing Activities

Product Critique:

- Testers pair together and exploratory test new builds
- Testers and developers pair together:
 - during development (to test specific areas of functionality)
 - after development (to run technical tests such as performance, data quality tests etc.)



Business Facing Activities

Business Support:

- Testers and customers pair together
- Testers act in a:
 - Support role (to develop test ideas and to help customers run through UAT)
 - Communications role (to help customers convey their impressions of the program to development)



Business Facing Activities

Product Critique:

- Testers exploratory test together
- Testers and developers pair together for black-box testing to:
 - ensure the iteration goals are met
 - generate test ideas
 - track down elusive defects



Pair Testing Benefits

Project Benefits:

- Testers are “information conduits”
 - Dedicated testers often have the most information to share between the business and the technical people
 - Good testers are effective communicators
- Reduction of duplication of effort
 - It is possible to have two or three groups developing the same tests and not sharing
 - Less “testing in the silo”



Pair Testing Benefits

Product Benefits:

- Broader testing activities:
 - Potential for more testing activities than automated unit tests and UAT
 - The more testing techniques used on a product, the better the risk mitigation
 - Increases the project focus on meeting the needs of the customer and producing a high-quality product



Benefits for Developers

- Developers (who are now doing much more testing) can draw on the skills and knowledge of dedicated testing specialists
- Skilled testers can quickly help developers find integration bugs and other problems that automated tests may not uncover
- 100% test automation is a good goal, but until we can program inference, it can't mitigate all the risks in a complex program



Benefits for Testers

- Testers can leverage the expertise of developers in test automation efforts (See James Bach's Test Toolsmith role in *Agile Test Automation*)
- Developers can share different testing techniques with testers
- Developers' unique view of the program can identify areas of risk that others may not see
- Testers learn new testing techniques from developers



Pair Testing Benefits

Testers also notice these benefits:

- Speed – a pair of experienced exploratory testers can find important bugs very quickly
- Enhanced improvisation, risk-taking and creativity
- Back up – potentially contentious issues can be expressed to the team better by a pair
- Perhaps most importantly, it is fun to pair test



Benefits for Agile Projects

Exploratory Testing and Agile Projects:

“Because an agile development project can accept new and unanticipated functionality so fast, it is impossible to reason out the consequences of every decision ahead of time. In other words, agile programs are more subject to unintended consequences of choices simply because choices happen so much faster.

This is where exploratory testing saves the day. Because the program always runs, it is always ready to be explored.”

- *Ward Cunningham, on Exploratory Testing*



Exploratory Testing Defined

- Simultaneous learning, test design and execution - *James Bach*
- For more, check out Bach's *Exploratory Testing Explained*:
www.satisfice.com/articles/et-article.pdf
- Agile methods and Exploratory Testing are a good fit



Exploratory Testing in Pairs

- Agile projects can benefit from Exploratory Testing activities:
 - Business Facing tests such as GUI-level Exploratory Testing
 - Technology Facing tests through an API or other published interface
 - See Brian Marick's article *Behind the Screens* in this month's Better Software magazine



Pair Testing on Agile Projects

A powerful combination of activities:

- Designing for testability (TDD and driving out testable interfaces)
- Automating tests and continuous integration
- Skilled Exploratory Testing (especially in pairs)



Pair Testing and TDD

Developer/Tester Pairing:

- Currently, there is not a lot of published work in this area (more needs to be done)
- Some TDD pairing activities are a good fit, others are more difficult



TDD Testing Activities

William Wake:

- There are two testing activities in TDD
 - Generative Tests
 - “shallow sweep” tests that yield design
 - not necessarily a natural fit for dedicated testers
 - Elaborative Tests
 - deeper, more detailed tests
 - play out the theme of the design and variations
 - a natural fit for testers



TDD and Pair Testing

What a dedicated tester needs to know:

- Start pairing in the elaborative test phase
 - Work towards the generative test pairing with a developer who enjoys the interruption of critical thinking when creating code
- Know the business rules that need to be tested before pairing
- Bring useful testing and development documents such as data dictionaries, story cards and use cases



TDD and Pair Testing

- Watch for testing smells – they will demonstrate bad coding smells
- Ensure that the interfaces the developer creates are testable
- Don't know how to watch for the above?
 - the developer will be happy to teach you
- When pairing during generative test development, a tester can:
 - act in a pair programming role
 - quietly take notes and think of test strategies
 - always be respectful when voicing testing concerns



Tips for getting started

- Define an area of testing focus:
 - track down an elusive bug
 - test new functionality at the end of an iteration
 - explore unpublished capabilities of the program
- Decide a time period of testing (an hour is a good place to start)
- See Kohl's *Pair Testing* article and Kaner and Bach's *Exploratory Testing in Pairs* presentation for details



Suggested duration of sessions

- Dedicated Tester pairs can probably pair as much as programmer pairs (the majority of the work day isn't uncommon)
- Tester/Developer pairs should start out with an hour or two a week and work up from there
 - many developers need a protected time of creativity when developing software
- Other pairing efforts can start at an hour or two after an iteration and work up from there as needed

References

- Wake, William. Email to presenter. March 2004
- Cunningham, Ward. Email to agile-testing list. March 2004
- Kaner & Bach. *Exploratory Testing in Pairs*. www.kaner.com/pdfs/exptest.pdf
- Marick. *Agile Testing Directions*. www.testing.com/cgi-bin/blog
- Bach. *Agile Test Automation*. www.satisfice.com/presentations/agileauto.pdf
- Marick. "Behind the Screens." *Better Software*. April 2004.