

# Micro-Pairing

(why arguing while you code is fun and productive)



**Brad Wilson**

<http://bradwilson.typepad.com/>

**Peter Provost**

<http://www.peterprovost.org/>

**Mitch Lacey**

<http://www.mitchlacey.com/>

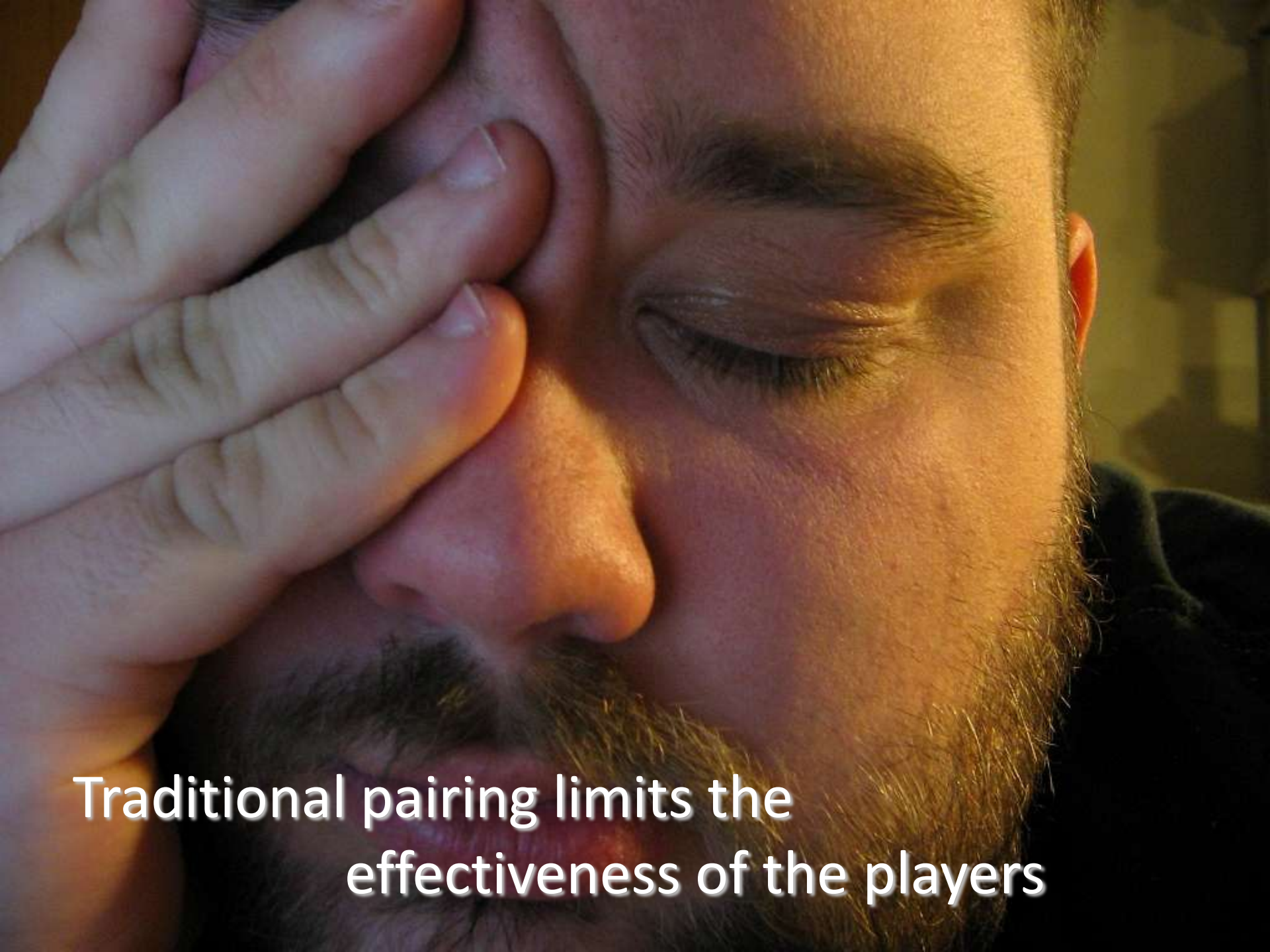
**Microsoft Corporation**

Teams today are adopting agile at a rapid rate





Developers want to make the best use of their coding time



Traditional pairing limits the effectiveness of the players

Successful pair development  
produces better designs



Change your pairing style to be a more effective, happier developer



# Pair Programming

A close-up, shallow depth-of-field photograph of a black computer keyboard. The keys are dark with white lettering. The focus is sharp on the 'D', 'F', 'C', and 'V' keys in the foreground, while the rest of the keyboard and the text 'Pair Programming' are slightly blurred. The text is centered at the top of the image in a clean, white, sans-serif font.



Two developers  
at one computer



The driver is  
the tactician




A photograph of a chessboard with glass chess pieces. The board is checkered, with light and dark squares. The pieces are made of clear glass and are arranged in a line across the board. The background is dark and out of focus.

The observer is  
the strategist

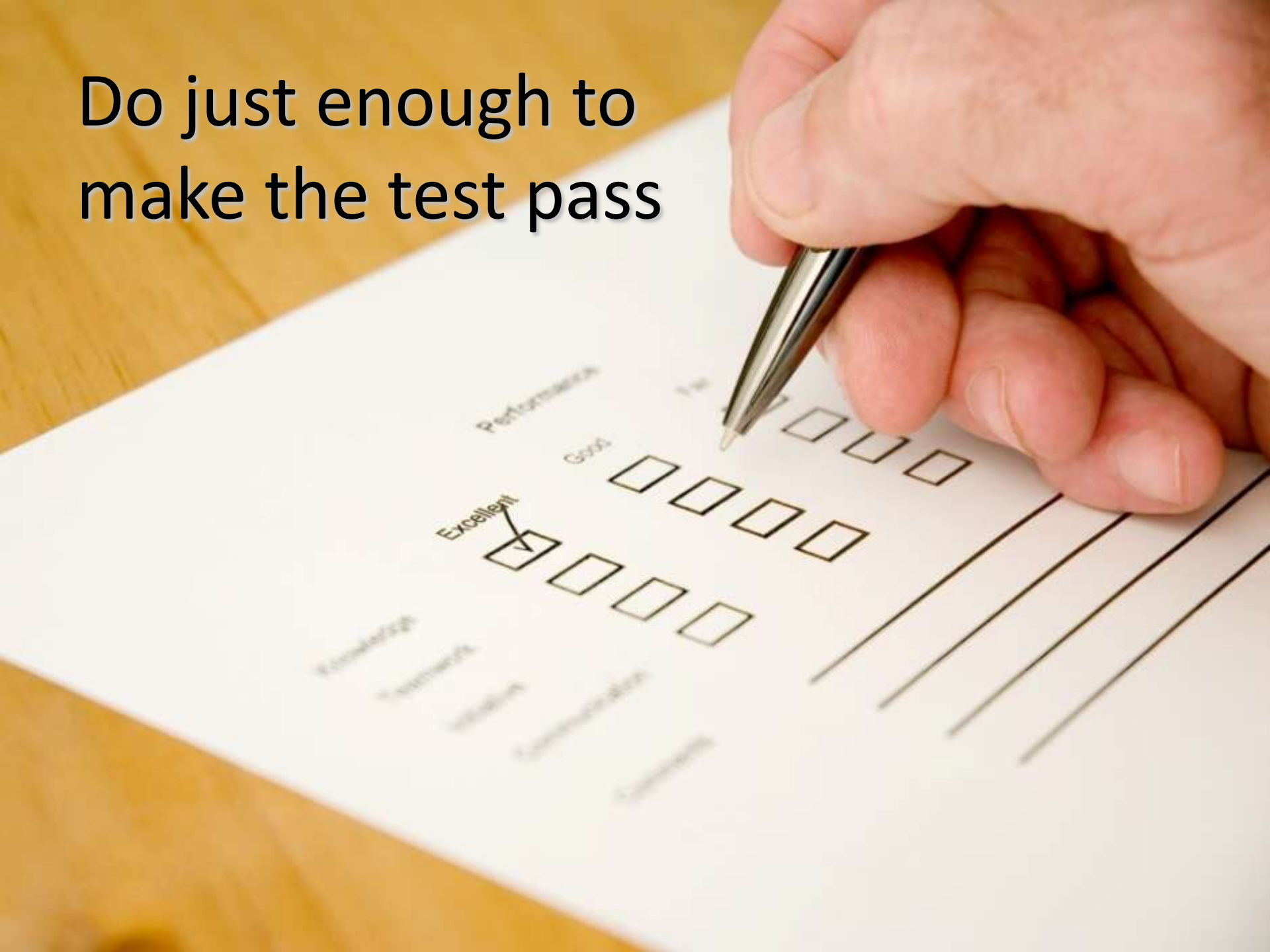


Test  
Driven  
Design



**Write a test  
before writing code**

Do just enough to  
make the test pass





Do It Together



Pass the keyboard  
at every  
TDD step







Negotiate over your disagreements

Have fun



Can you learn to become a more effective pair in just 90 minutes?



Developers become bored and ineffective  
with traditional pair-programming





Adjust your style

to increase productivity, design quality, and fun

```
i = x
```

```
while( n < (docum  
{
```

```
n++;  
calc = ev  
i++  
i++
```

Become a more  
effective developer  
with the game



# The Pairing Game

Bringing fun to agile  
software development