Pair Programming

If you have ever worked in a development role, chances are ~~that~~ you are familiar with ~~the term~~ pair programming **-** ~~and~~ you may even have first hand experience. But whether you were an instant fan or a skeptic, **pair programming** ~~programming in pairs is a concept that~~ is here to stay. With this in mind, I would like to take a few minutes to discuss pair programming as well as some best practices that will hopefully help you make the most out of your next encounter with this controversial phenomenon.

The Theory

The ~~theory behind~~ argument for pairing is that it can ~~will~~ result in faster programming, more creative solutions, fewer bugs, and ~~perhaps~~ most importantly ~~of all~~, ~~it~~ spreads the knowledge of your code base across the team. Arguably one of the biggest issues in a team is having one developer who knows the code base better than any other~~,~~ This can burn out the single developer who ~~the one that~~ gets all the weekend calls and is expected to have all the answers. When you have more than one developer working on a section of code, you invest the time to spread the knowledge amungst your team.

The Mechanics

The concept of pair programming revolves around two developers working together on one terminal. The pair can be broken down into two major roles, the driver and the navigator. The “driver” operates the keyboard and is responsible for physically ~~recording~~ **inputting** the code, while the “navigator” focuses on context and the overall direction. Roles are switched frequently throughout a session with the overall goal being that each person spends an equal amount of time in each role.

Success in Pairs

What is it that defines a successful partnership? Aside from the obvious~~ness~~ requirements of ~~a~~ good attitude and personal hygiene, a developer can ~~what can you~~ bring the below to this unique partnership to ensure its' success.

•**Think out loud.** Something as simple as vocalizing your thoughts allows your partner to work with you on the same level, guide you when you are stuck and even bring you to an entirely new path if required.

•**Discuss role expections and programming styles.** If your aren't familiar with your partners coding style, you can't make the most out of your time as a navigator.

•**Have a development plan.** Try pairing pair programming with your favorite development style~~,~~. An~~an~~ example from test driven development would be to switch the partners between ~~one~~ writing a test and ~~the second~~ programming to pass the test.

Conclusion

Although not all situations call for pair programming, it ~~is~~ can a great tool to improve teamwork and promote an understanding of the code base in its' entirety. Just keep in mind that we are humans and when forced upon those who are truly uninterested, the outcome is often less than fruitful.