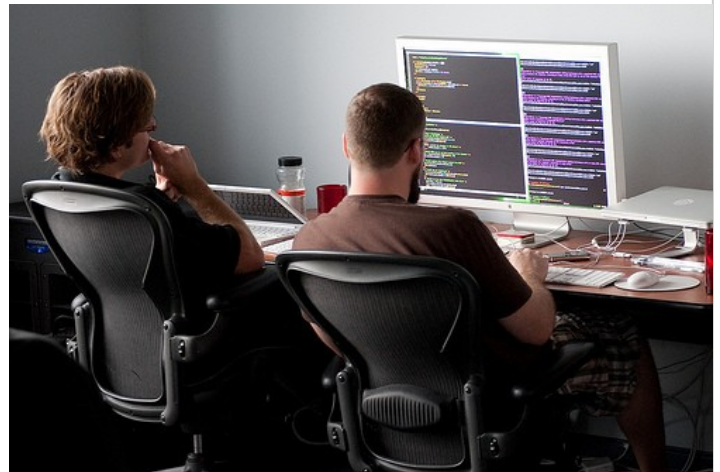


Pair Programming

- Pair Programming is one of the core practices of extreme programming and literally means that all production software is written by two programmers, sitting side by side, at the same machine. This practice ensures that all production code is reviewed by at least one other programmer, and results in better design, better testing, and better code.
- The actual practice of pairing requires programmers to frequently (daily or multiple times per day) swap pairs to promote sharing of knowledge throughout the team. As pairs switch, everyone gets the benefits of everyone's specialised knowledge.
- Contrary to popular belief that having two programmers doing "one programmer's job" is highly inefficient there's evidence that suggests that pairing produces better code in about the same time as programmers working singly.
- Some programmers object to pair programming without ever trying it. It takes practice and discipline to do it well, and you need to do it for a few weeks to be able to see results and appreciate the benefits.
- Like many other practices applying it blindly without using common sense is not recommended.



Things to consider:

- Maturity and experience of the team—more mature and experienced teams tend to have a faster uptake.
- Complexity of the code—many teams practice pairing for complex tasks and split up for ones that are considered straight forward.
- Time-to-market (TTM) and post release lifecycle — Faster TTM might be worth the higher cost of development in exchange for better quality code that will pay off over time.



REFERENCES

- <http://xprogramming.com/xpmag/whatisxp>
- <http://agilesoftwaredevelopment.com/blog/artem/pair-programming-what-researches-say>