WHAT IS PAIR PROGRAMMING?

Agile software development is the organization of people to work independently, make their own decisions, and contribute to the greater whole. Of course, the overarching purpose of agile development is to allow teams to adapt quicker to the demands of the market.

Pair programming has emerged as a useful tool in the agile software development toolbox.

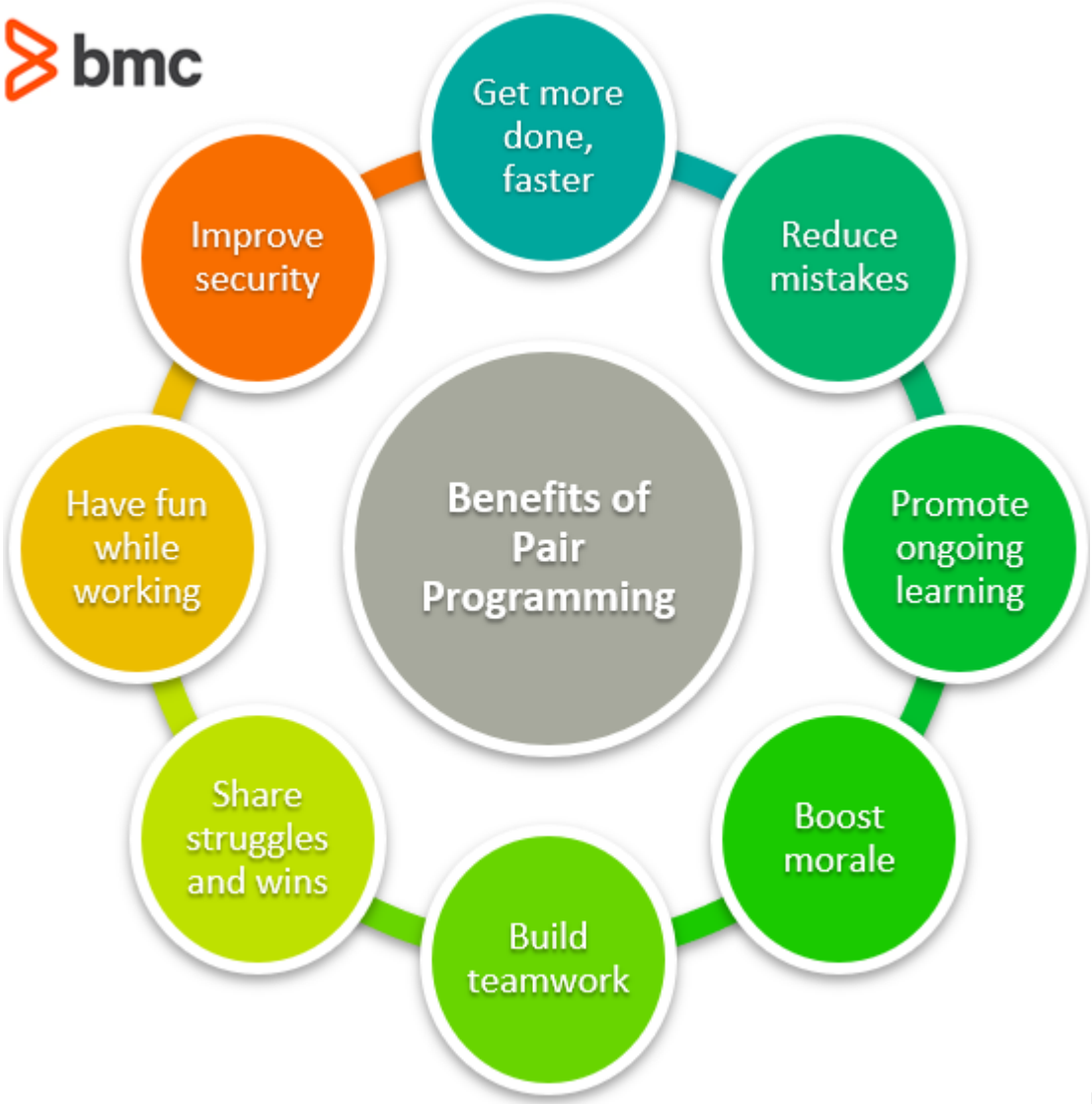
Drawbacks of working alone

Traditionally, programmers work alone, building a feature or function or a single app. But that's not always ideal. Working by oneself can be monotonous but the reverse—working with a large group of people—limits your individual autonomy. In groups larger than 4, one person's intelligence tends to dominate.

So, working in a pair, for many programmers, is just right.

Pair programming: two heads are better

Pair programming is intentionally putting together two developers to accomplish a task. Both the business managers and the devs themselves appreciate this two-person approach, but often for different reasons.



For the business managers with their eyes on KPIs, they like to think that pair programming increases productivity. Pair programming:

- Reduces mistakes
- Gets more done, faster
- Allows the company to act when problems arise
- Boosts programmers' morale

For the programmers, pair programming:

- Lets you talk with a partner
- Allows you to share your struggles and accomplishments when solving a problem
- Promotes continuous cross-training
- Builds team and coder relationships
- Encourages both learning and teaching, increasing your sphere of value

For the coder, pair programming can feel like you're sticking it to the man, giving the pair a sense of purpose. Because, while the company leaders believe this work practice increases productivity, which may be true, the coder gets to have more fun.

How pair programming works

Pair programming is a practice where two programmers work together. Often one will write code while another reviews the code. They go back-and-forth with their roles, much in the same way as British Primary Pedagogy (multi-age classrooms), getting to be both the player and the coach. Both the writer and the reviewer. Or, in the words of the pair, the driver and the navigator.

The pairing method is great for learning. Developers who program in pairs report they:

- Learn faster
- Make less mistakes
- Spend less time on small problems and more time on large problems

While the driver gets to show off their talents, or just get personal attention from another to learn, in a pair, each person gets to feel special. How and when breaks are taken are up to the pair. Improvise; trust the instincts. When the trust is lost, research has been done, and some swear by the Pomodoro Technique or the 52/17 method to define their work-to-break ratio.

Pair programming increases security

Perhaps the single largest benefit of programming in pairs is that it increases the security of whatever you're building. The resulting product is likes to have security benefits like:

- Fewer bugs
- Internal auditing against from backdoors

Fewer bugs

Through pair programming, an extra set of eyes catches potential bugs in the code. We know when code works, it's excellent. But code can work for many, many reasons. In those ways the code can work, it can perform its task in unsuspecting ways, too. When code works the way a developer wants it to, but has many methods of working, or works for the wrong reasons, this becomes a vulnerability for adversaries.

Cheats, or hacks, are made. If Mom said, "Go to bed at 8pm," and her only verification for telling time were the clocks in the house, then it's possible to change all the clocks in the house so bedtime is later. When multiple eyes look at code, they can catch simple bugs, and write better code to prevent easy hacks before they occur.

Minimal backdoors

While each pair reviews each other's' work, they can also notice if the other has built a backdoor into the software. Unless they're a George Clooney and Brad Pitt duo, pair programming has a built-in security method society calls accountability, or integrity, that ensures the submitted code is clean.

When to use pair programming

Not every task or project is well suited for pair programming.

Programming in pairs is great for training. The back and forth dialogue helps the pair understand a

new concept or where another is coming from. It builds teams, and trust among teams. Pair programming, then, is ideal for:

- Teaching students
- Training new staff
- Solving cross-functional problems
- Promoting continuous cross-functional training
- Ideating and innovating

But, when the task is ahead, it is well-defined, the sprint needs to be made, and everyone knows what they're doing, sometimes it is the good old-fashioned headphones-on, coffee-at-the-ready grind that a programmer needs to do to get the job done.

Additional resources

For more on this topic, check out the BMC DevOps Blog or read these articles:

- The Software Development Lifecycle (SDLC): An Introduction
- The Role of Agile in DevOps, part of our DevOps Guide
- Scrum vs Kanban: A Comparison of Agile Methodologies
- Application Developer Roles and Responsibilities
- What is a Citizen Developer?