Pair Programming

An Introduction to Pair Programming

Pair programming in JavaScript is a collaborative software development technique where two programmers work together on the same task. First and foremost, this technique promotes better code quality. Having two sets of eyes on the code helps catch errors, bugs, and logical flaws early in the development process.

Another advantage of pair programming in JavaScript is knowledge sharing. Each programmer brings their unique skills, experiences, and perspectives to the table. By working together, they can learn from each other, exchange ideas, and expand their knowledge base. This collaborative learning environment fosters professional growth and helps build a stronger team.

In addition, pair programming in JavaScript helps reduce the number of bugs and issues in the codebase. With two programmers actively involved in the development process, the chances of overlooking errors decrease significantly. The navigator's role in reviewing the code and providing feedback ensures that potential issues are caught early, leading to a more robust and stable codebase.

Pair Programming Styles

Driver-Navigator

In the most common pair programming style, driver-navigator, each developer has their own monitor, mouse, and keyboard. Each monitor is a mirror of the other. In this way, both developers can sit comfortably, have a full view of the screen, and take or relinquish control as needed. When two developers are working remotely, they can use a tool for remote screen sharing such as the screen sharing feature in Slack.

Pair programming is a mutual activity, and it won't be as effective if only one developer is always the driver. Ideally both the team members should periodically trade roles for better results. The driver writes the code while the navigator observes and thinks ahead, asking questions and offering immediate suggestions for improvement. The navigator is essentially doing a real-time code review as the code is being written.

Strong-Style

Strong-style pairing is the approach in which the driver doesn’t do anything that the navigator did not tell them to. Whenever the driver would like to pitch an idea, the driver must then handover the system to the navigator and then effectively swap positions. This approach completely engages each programmer.

Tour Guide

In the tour guide style, the driver acts like a tour guide on a bus. The driver does both the strategic thinking and the typing. The other pair member is the "tourist" who passively listens. The tour guide model is useful when one person has much more context than the other, but it's not an ideal long-term arrangement. Even with a significant skill gap between pair programmers, learning by doing is more effective than learning by watching.

Ping-pong

Ping-pong pairing is another pair programming pattern frequently employed by developers. Ping-pong programming works well when combined with test-driven development. In the ping-pong style, one member of the pair writes the test then the other member of the pair writes the code to make the test past. That pair member then writes the next test before passing control back to the other. The pair continues this way throughout the development.

Pomodoro

The Pomodoro pairing style is like ping-pong pairing but prescribes set time intervals for each session. A typical Pomodoro-style pairing session lasts 25 minutes followed by a 5-minute break. The driver and navigator then switch positions. After four 25-minute sessions, both programmers take a longer 20-minute break. Forced breaks and regular position switching help ensure that both programmers are always productive, focused, and refreshed when a session begins.

Tag-Team

In tag-team pairing, the driver and navigator switch positions but don't follow set time intervals or rules for when the switching can occur. Basically, the switch can occur when the driver is tired, when the navigator wants a chance to take the lead, or for any other reason given by either programmer.

In Conclusion

Pair programming in JavaScript offers numerous benefits. It promotes better code quality, knowledge sharing, productivity, problem-solving skills, team dynamics, bug reduction, and learning opportunities. By leveraging the strengths of two programmers, this collaborative approach leads to more efficient and effective software development. Whether you're a beginner or an experienced developer, pair programming in JavaScript can greatly enhance your skills and contribute to the success of your projects.

Author: Trevor Morin