

# **Improving Code Quality, Velocity, and Team Cohesion with Pair Programming**

Demethria Ramseur, Ph.D

<https://www.linkedin.com/in/demethriaramseur/>

## **ABSTRACT**

Working in a globally distributed engineering team comes with a unique set of challenges. On the technical side, it is challenging to balance development velocity and code quality. On the team-building side, it is hard to create a sense of camaraderie when people only encounter each other in periodic video-based scrum calls. This paper focuses on the Agile practice of pair programming that can be applied to improve the overall level of code quality, increase coding velocity, and to help build relationships.

## **INTRODUCTION**

Common challenges remote and distributed engineers face are maintaining productivity and velocity, balancing speed and quality, and working as a team. Many of the challenges stem from a lack of collaboration. A collaborative team culture is important for transparency, productivity, and reliability. Pair programming is a software development approach that improves code quality, increases velocity, and improves team cohesion. This paper covers pair programming benefits, pair programming misconceptions, and factors to consider when adopting pair programming for a team or organization.

## **PAIR PROGRAMMING DEFINED**

Pair programming is an Agile software development technique where two or more developers work together to solve a problem. Developers share ideas real time and develop code synchronously. One of the developers, referred to as the driver, has their hands-on key-board and types the code while explaining their thought process. The other developer, referred to as the navigator, watches the screen, reviews the code, raises any concerns, helps in error identification, and collaborates on the development effort. The developers swap roles periodically throughout the pairing session.

Pair programming is not a new concept. The pair programming concept originates from extreme programming that has been around since 1996. In extreme programming, two developers must always work on production code together. By pairing, the developers are able to collaborate and share ideas which leads to early identification of bugs and other issues that could impact coding velocity and code quality.

## PAIR PROGRAMMING BENEFITS

In pair programming, there is another developer to bounce ideas off of or to obtain another perspective on how to approach solving a problem. The collaboration that occurs when pair programming leads to benefits such as increased velocity, improved code quality, team cohesion, and provided mentoring opportunities. This section covers pair programming benefits and how they are achieved.

### Increased Velocity

One benefit that can be achieved with pair programming is improving the ability to complete work and/or delivery features faster. Velocity can be increased by getting unblocked faster, getting code changes approved faster, and identifying code errors faster.

***Getting unblocked faster:*** Some developers will work independently for longer than needed amounts of time on trying to resolve code issues instead of setting shortened time periods for working through issues and getting help when the time period ends. This leads to asking for help later versus sooner, resulting in a missed opportunity to complete work by the requested date. Sometimes a second set of eyes is all that is needed to get unblocked, resolve an issue, and move work forward. Pair programming allows another developer (that second pair of eyes) to be a part of the coding session to help discuss any issues and move work forward. Pair programming makes it easier for developers to discuss issues and accelerate the ability to complete the work, because via the pairing session, the help needed to get unblocked is provided.

***Getting code changes approved faster:*** Since two developers are working on the code together, both developers will be familiar with the code. Code changes submitted during the pairing session by the developer acting as the “driver” can be approved during the session by the developer acting as “navigator” and visa-versa. Code changes submitted after a pair programming session can also be approved faster since there are two (or more) developers familiar with the code. A developer who is familiar with the code and understands the context of the change will be able to approve the change faster than a developer who is unfamiliar with the change and context, resulting in a faster approval of the change.

***Identifying code errors faster:*** While the developer acting as the “driver” types, the developer acting as the “navigator” is watching and reviewing. The navigator is able to identify typos, incorrect use of functions, and other coding errors in real time so they can be corrected. Finding the coding errors while the code is being developed alleviates the time that would be taken to troubleshoot failing code and finding out it was due to a coding error.

### Improved Code Quality

Improved code quality is another benefit that can be achieved by pair programming. Code can be improved by sharing ideas and making efficient code.

***Shared ideas:*** Developing features (aka coding) is more than just typing the code. Development requires a design and/or an algorithm; therefore, a plan of approach is needed. During pair programming sessions, there are two or more developers working together and sharing ideas and different perspectives on how to solve a problem. The discussions and sharing of ideas allow for better or different plans of approach to be revealed. The idea sharing also helps developers think through use cases that may not have been considered.

***Make code efficient:*** During pair programming, the “navigator” may suggest ways of writing the code to make it more efficient while having the same outcome.

## **Build Team Cohesion**

With teams being distributed and team working remotely, some developers feel isolated when working alone. Pair programming helps to overcome the feelings of isolation by providing an opportunity to learn about a teammate during the pairing session and by helping the team feel more connected. Even though pairing implies two, more than two developers can participate in a pairing session. Other team members can be invited to join the pairing session to listen in. Others can learn and develop by hearing how others solve problems or think through how to address issues.

***Learning about your teammates:*** The pair pairing sessions allow the developers participating in the session to learn each other's strengths and weaknesses. The pairing session provides an opportunity for the developers pairing to see what tools are used or what commands are used. Sometimes subtle ways of working may be uncovered that no one would have thought to ask about. Pair pairing helps with getting team members to become more comfortable working together and helping each other more. Pair programming helps build relationships within the team.

***Helps the team feel more connected:*** Pair programming helps all developers on a team feel more connected when they are able to collaborate on feature development together. By having one driver and multiple navigators, the whole team can share ideas and perspectives and contribute to a development item. The whole team feels connected because they all contributed to the success of the development item.

## **Mentoring Opportunity**

Pair programming can be leveraged as a mentoring opportunity by senior developers or by anyone on the team that has knowledge and experience that they want to share. Senior developers can ask junior team members if there is anything they want to pair on. Senior developers can also invite other team members to pair with them on work items.

## **Summary**

Pair programming can improve code quality through continuous review, reduce bugs through real-time error identification, and facilitate knowledge sharing among team members. Moreover, pair programming can alleviate the feeling of isolation that some developers experience when working alone.

## **PAIR PROGRAMMING MISCONCEPTIONS**

The Merriam-webster dictionary defines misconception as “a wrong or inaccurate idea or [conception](#)”. Pair programming has many benefits; however, there are misconceptions that make some developers reluctant or hesitant to try pair programming. Let’s explore some pair programming misconceptions.

### **Only for Junior Developers**

One misconception is that pair programming is not for experienced developers. That pair programming only is helpful for Junior developers implies that senior developers will not benefit from pair programming.

Pair programming helps to surface different perspectives and insights and developers of all skill levels can benefit from sharing and collaborating.

### **Results in Longer Completion Time**

A common pair programming misconception is that it slows down the ability to complete development work within the requested or expected time frame. Developers believe that working in pairs slows down development or development speed and giving someone context to be able to contribute can take a long time.

Although taking longer is a misconception, pair programming results in increased velocity.

### **No One Has Time**

Another misconception is that other developers do not have time or won’t make the time to participate in pair programming because they have their own work to work on. Not wanting to tie up another engineer for a long stretch of time may make a developer reluctant to try pair programming.

Pairing sessions don’t have to be long to be beneficial. 1-2 hours tops is sometimes all that is needed for the session to be effective. Furthermore, people like to help others and if given the opportunity, developers that think individuals do not not have time, will see individuals make time if presented with the opportunity.

### **No “I did it myself” Feeling of Accomplishment**

Some developers may be reluctant to try pair programming because they do not believe they will have the same feeling of self accomplishments. They believe solving the problem by themselves is the only way to realize a sense of accomplishment or they view pairing as not proving their own abilities and making their own achievements.

Feelings of accomplishment are felt when helping others achieve their goals and when

someone helps you achieve a goal. There is a feeling of accomplishment even when pair programming.

### **Mismatched Skill Levels**

Some developers may be reluctant to pair due worrying about being judged or not being able to contribute equally. Some developers may lack confidence in their programming skills or may have concerns that working with developers with different coding styles may result in conflict.

Pair programming helps to overcome these misconceptions. Pair programming is meant to share different ideas and perspectives. Everyone doesn't have to be at the same level to contribute and for pairing to be effective.

### **Summary**

The misconceptions individuals have about pair programming are very normal. These misconceptions can be overcome by trying pair programming and experiencing the benefits.

### **PUTTING PAIR PROGRAMMING INTO PRACTICE**

Pair programming has many benefits as discussed in this paper. Items to consider when starting pair programming are discussed in this section.

#### **Tools**

In order to perform pair programming, a screen sharing program, such as google meet, is needed to allow the developers to see each other's screen. The code must also be in a common location that both developers can access such as github.

#### **Guidance**

A set of guidelines should be agreed upon by the developers that will be pairing. A couple of items to consider when determining the guidelines are the length of the pair programming session and at what point the "driver" and "navigator" should swap roles.

Sharing a screen for long periods of time may result in screen fatigue. A good rule of thumb is to keep the sessions to 1-2 hours.

Throughout the session, the "driver" and "navigator" should swap roles. A general guide is every 30 minutes; however, the developers can decide what points are appropriate for swapping.

#### **Stakeholder Support**

If pair programming is being rolled out to a team or larger organization, obtaining

stakeholder support will be beneficial. The stakeholder(s) will help to promote the idea of pair programming with the team and/or organization by discussing the benefits of pair programming in meetings and promoting the idea in other forums. Having stakeholder support will help to increase the likelihood of the developers participating in pair programming.

### **Pair Programming Champion**

The pair programming champion is an individual who believes in the benefits of pair programming and promotes the idea of pair programming within the team and or organization. The pair programming champion sets the example for how pair programming should work by inviting individuals to pair on topics and by asking others if there are things that they want to pair on. Pairing with the champion helps others experience the benefits of pair programming.

### **Experiment**

Experimenting with pair programming is necessary to determine which approaches work best for the team or organization. Three approaches that can be tried are as follows:

- Assign two or more developers to implement a work item
- Assign each developer to a work item and challenge each developer to pair with one person from the team
- Have the whole team work on a large work item together. In the whole team pairing approach This approach pair where the entire team worked on an epic (development for the same feature). Pairs or more have a dedicated set of meetings to work on specific tasks with one driver and one or more others to discuss

### **Continuously Improve**

Continuously improve the pair programming approach being performed by the team. Evaluate which pair programming approaches the team likes or dislikes. Determine whether more guidelines are needed? Determine whether any processes are needed to support the pair programming within the team or organization. There may be components of pair programming that require more formalization. There is no one size fits all approach. The best approach is the approach that works best for you, your team, or your organization.

### **Team Culture**

In order for pair programming to be most effective, developers need to be comfortable sharing ideas and not be judged based on their comments and feedback. There needs to be a blameless culture and everyone should assume positive intent. Pair programming is suited for environments with a A blameless culture or a culture where everyone assumes positive intent

### **CONCLUSION**

In conclusion, and as you contemplate the use of pair programming, here are a few final points to consider. Pair Programming has numerous benefits such as increased velocity, improved code quality, and improved team collaboration. There is no one size fits all when it comes to pair programming and what works for different individuals and teams. Pair programming requires a

little practice and iteration. Try it and determine what works best for you, your team, and/or your organization.

## REFERENCES

Agile Alliance. “AGILE GLOSSARY: Extreme Programming.” Accessed Feb 18, 2024.

<https://www.agilealliance.org/glossary/xp/>

Catalino, A. (2021, Dec. 2). “7 Myths of Pair Programming”. *Drovio*.

<https://www.drovio.com/blog/7-myths-of-pair-programming/>

Sharovатов, V. (2023, Oct, 17). “A guide to pair programming: a top software development method”. *Qase*. <https://qase.io/blog/pair-programming/>

“What Are Some Benefits of Pair Programming?”. Accessed Feb 18, 2024.

<https://makemeaprogrammer.com/what-are-some-benefits-of-pair-programming/>