

# Creating A Culture Of Mentorship

## Introduction

Many teams are looking for ways to foster a culture of mentorship, but other than 1-on-1 mentorship sessions, they aren't sure how to go about accomplishing it. There are a lot of ways to improve the culture of mentorship on your team. Overall, the goals are to create a general hunger amongst the team to learn new concepts and then share that learning with others. This excitement, drive for self-improvement, and the desire to share those improvements with a team are the hallmarks of successful mentorship-oriented teams.

Here are some activities that we've seen on team with great mentorship cultures:

## Formal Mentorship

This is the most obvious and common form of mentorship on teams. The team splits into pairs of senior and more-junior developers, go into a room, and talk about development topics. Even though this is obvious and common, it shouldn't be taken to mean that it's not valuable. Developers in active mentorship relationships (both mentors and mentees) are significantly more likely to be promoted than their peers. 1-on-1 mentorship creates an environment for developers to get feedback and training tailored to their specific needs and interests.

If your team is not already doing formal mentorship, then start here.

## Code Reviews

Two mistakes that people make regarding code reviews are: (1) only having the tech lead or other senior engineer do all the code reviews, and (2) treating code reviews as activities with no mentorship value that need to be finished as quickly as possible. Code reviews have a lot of positive mentorship benefits, including:

- Code authors learn from reviewers through the process of accepting feedback
- Reviewers learn from code authors by seeing how the author tackled challenges
- Reviewers learn about code they might be unfamiliar with through the process of review. This reduces the criticality of any one team member by spreading their knowledge across the team
- Authors and reviewers learn by sharing refactoring ideas or having discussions about alternative approaches
- Authors and reviewers learn how to negotiate and manage conflict constructively

Because of this, all members of a team (regardless of how junior or inexperienced) should be performing code reviews. If necessary, have more than one reviewer on a review to make sure that nothing important slips through. Make sure that the team has enough time to do reviews correctly (it can take 1hr or more to deeply review 100 lines of code). Lastly, make sure that team members understand the importance of code reviews. Once all members of the team feel responsible for the quality of the application, the level of code reviews rises dramatically.

## Pair Programming

Pair programming is one of the more controversial mentorship activities. When done effectively, it is one of the quickest ways to share knowledge between developers and coach-up junior developers. To maximize the mentorship value, pair juniors with senior developers. Let the seniors dictate what is to be done and let the juniors “drive”. Make sure that the senior is allowing the junior to be involved in not only understanding the solution but also in determining how to implement it.

It's important to make sure that pair programming is done sparingly in accordance with the preferences, talents, and makeup of your team. Be mindful of power imbalances between the paired programmers: juniors and seniors, men and women, minoritized groups and others, assertive personalities and agreeable ones, etc.. Train the team for how to be aware of these imbalances, how to address them, and how to handle conflict.

Pair programming doesn't have to be completely formal, either. If you're a team leader, it's often enough to make room for and promote informal pairing, where team members request the assistance of others for an extended period to address a particular issue. This might be less efficient in the short term, but it will pay off in the future.

## Tech Talks

Tech talks are a great way to share knowledge, build team culture, and promote the professional development of your team members. A tech talk is when a developer prepares a conference talk-like presentation and then delivers it to their peers. The audience learns new information from the presenter via the talk. The presenters get to practice communication skills and are introduced to the potential joys of public speaking in a relatively friendly, controlled atmosphere.

These talks can take place over lunch, at special one-off events, or during scheduled team meetings. The talks can vary in length from 5 minute lightning talks to 20 minute quick dives to longer 40 minute long talks.

Many conference circuit speakers started their speaking careers doing tech talks at their companies and local meetups.

# Chat Channels

An often overlooked source of casual mentorship in a team are the chat channels or forums that a team might use to communicate. You want to create a place where your developers can share links, discuss problems and solutions, and just generally have conversations about development. You want this to be casual. People should feel free to jump in and jump out of conversations based on how available they are and how relevant the conversation is.

Make sure that you give people enough structure so they can find the topics they're interested in without getting overwhelmed by all the random chatter. You can make topics for programming, but also topics for hobbies such as cooking, parenting, workouts, music, etc.

Not only do the chat channels promote learning and sharing what you've learned with others, but they also help promote a healthy team culture. This is especially powerful in teams that are mostly or completely remote.

# Conclusion

A team with a strong culture of mentorship is full of developers that are excited to go to work, excited to learn and improve their development skills, and excited to share what they've learned with their coworkers. The environment of skills development and shared excitement leads to a dynamic workplace with high engagement, low burnout, high retention, and great career advancement. To create this culture, it's important to move beyond simply holding formal 1-on-1 mentorship sessions. Instead, creating multiple different spaces where developers can collaborate and learn from each other.

Make sure that management, senior developers, and team leaders lead by example and show that these activities are important and worth the time spent on them. Once the team understands that these activities are valued, that there is an investment of time and resources to support these activities, and that there are a variety of ways to be engaged, they will start to participate freely.