ADJUSTING TO CHANGE IN YOUR AGILE TEAMS

It's been widely reported that we are in the middle of the great resignation, and our teams could be facing a shake-up, with old members leaving and new members arriving to replace them. While we appreciate a smoothly running, stable team, we must also adjust to this reality and turn it to our advantage. How do we keep the rhythm of a high-producing team going when the members change?

When we bring in new developers, our primary focus is often on having them master the application, but in focusing so heavily on that, we can often overlook their need to master working with the team, too. Even when teams all follow the same agile rules, they all work differently—you can master the code but still fail because you haven't mastered the team.

Scrum master Arshind Kaur offers the following advice, "Just like with deliverables, establish an MVP for new hire expectations. It is important to ensure that their training/team interactions help them understand the value of what we are doing and our mindset while we do it. The clerical or more detail-oriented practices can come later. For example, if a team member is new to JIRA/Scrum practices of burndown or sub-tasking, give them time to understand and observe how this type of sharing and transparency helps the team to offer their help/ mentorship to each other, as well as allow other stakeholders to plan/remove impediments efficiently. Once the value of these practices is understood, the clerical aspects of JIRA, etc. can follow easily."

In addition to adding their development hours to the team, here are some ways new team members can also change teams for the better:

- **New perspectiv**e—New members bring in not only new skills but new perspectives. Be open to listening to them. Give them support during standup and refinement. Encourage them to provide their insight in a safe space. During retrospectives, make a point of asking them how they might have handled this situation in the past, what worked, and what didn't.
- **New mentors—**Getting new members can revitalize the whole team. For example, someone who was only recently a new member now has the opportunity to become a mentor and take more of a leadership role.
- **New development**—Learning how the team functions is as important as learning the code. Traditionally, new team members work on bugs to familiarize themselves with the many different parts of the application and see how it works and fails. After the first few bugs are successfully resolved, we put them on smaller, simpler, new development items to get them used to how we run our agile sprints, which also allows them to participate in planning, see how we handle new code, go through code review, work in a two-week sprint, and share their efforts in the sprint review.
- **New skills**—Producing a well-rounded developer helps everyone. A developer's skills are shaped by the work that is available to them in their teams, so look for skills they have but might not been able to use on their previous team. For example, a developer may have great SQL coding skills that weren't needed on their previous team. If your team has this weakness, give the developer the opportunity to dive into an area that would benefit the team, and allow them to build on a dormant skill.

By focusing on the whole team and not just the new members, you will be able to navigate uncertainty. Accept the challenge of onboarding new members as an exciting opportunity for reinvention that pays off for the whole team.