Why Collaboration

Why is Collaboration Important

In engagements, we aim to be highly collaborative because when we code together, we perform better, have a higher sprint velocity, and have a greater degree of knowledge sharing across the team.

There are two common patterns we use for collaboration: Pairing and swarming.

Pair programming ("pairing") - two software engineers assigned to, and working on, one shared story at a time during the sprint. The Dev Lead assigns a user story to two engineers – one primary engineer (story owner) and one secondary engineer (pairing assignee).

Swarm programming ("swarming") - three or more software engineers collaborating on a high-priority item to bring it to completion.

How to Pair Program

As mentioned, every story is intentionally assigned to a pair. The pairing assignee may be in the process of upskilling, nevertheless, they are equal partners in the development effort. Below are some general guidelines for pairing:

- Upon assignment of the story/product backlog item (PBI), the pair needs to be deliberate about defining how to work together and have a firm definition of the work to be completed. This information should be expressed clearly in the story's description and acceptance criteria. The expectations about this need to be communicated and agreed upon by both engineers and should be done prior to any actual working sessions.
- The story owner and pairing assignee do not merely split the work up and sync regularly – they actively work together on the same tasks, and might share their screens via a Teams online session. Collaborative tools like VS Live Share can be preferable to sharing screens. Not all collaboration needs to be screen-share based.

- During the collaborative sessions, one engineer provides the development environment while the other actively views and comments verbally.
- Engineers trade places often from one session to the next so that everyone has time in control of the keyboard.
- Engineers leverage feature branches for the collaboration during the development of each story to have small Pull Requests (PRs) (as opposed to a single giant PR) at the end of the sprint.
- Code is committed to the repository by both members of the assigned pair where and when it makes sense as tasks were completed.
- The pairing assignee is the voice representing the pair during the daily standup while being supported by the story owner.
- Having the names of both individuals (owner and pair assignee) visible on the PBI can be helpful during sprint ceremonies and lead to greater accountability by the pairing assignee. An example of this using Azure DevOps cards can be found here.

Why Pair Programming Helps Collaboration

Pair programming helps collaboration because both engineers share equal responsibility for bringing the story to completion. This is a mutually beneficial exercise because, while the story owner often has more experience to lean on, the pairing assignee brings a fresh view that is unclouded by repetition.

Some other benefits include:

- Fewer defects and increased accountability. Having two sets of eyes allows the engineers more opportunity to catch errors and to remember often-overlooked tasks such as writing unit and integration tests.
- Pairing allows engineers with different experience and expertise to learn from one another by collaborating and receiving feedback in real-time. Instead of having an engineer work alone on a task for long hours and hit an isolation breaking point, pairing allows the pair to check in with one another.
- Even something as simple as describing the problem out loud can help uncover issues or bugs in the code.
- Pairing can help brainstorming as well as validating details such as making the variable names consistent.

When to Swarm Program

It is important to know that not every PBI needs to use swarming. Some sprints may not even warrant swarming at all. Swarm when:

- The work is complex enough to have collective minds collaborating (not because the quantity of work is more than what would be completed in one sprint).
- The task that the swarm works on has become (or is in imminent danger of becoming) a blocker to other stories.
- An unknown is discovered that needs a collaborative effort to form a decision on how to move forward. The collective knowledge and expertise help move the story forward more quickly and ultimately produced better quality code.
- A conflict or unresolved difference of opinion arises during a pairing session. Promote the work to become a swarming session to help resolve the conflict.

How to Swarm Program

As soon the pair finds out that the PBI will warrant swarming, the pair brings it up to the rest of the team (via parking lot during stand-up or asynchronously). Members of the team agree or volunteer to assist.

- The story owner (or pairing assignee) sends Teams call invite to the interested parties. This allows the swarm to have dedicated focus time by blocking time in calendars.
- During a swarming session, an engineer can branch out if there is something that needs to be handled while the swarm tackles the main problem at hand, then reconnects and reports back. This allows the swarm to focus on a core aspect and to be all on the same page.
- The Teams call is repeated until resolution is found or alternative path forward is formulated.

Why Swarm Programming Helps Collaboration

- Swarming allows the collective knowledge and expertise of the team to come together in a focused and unified way.
- Not only does swarming help close out the item faster, but it also helps the team understand each other's strengths and weaknesses.
- Allows the team to build a higher level of trust and work as a cohesive unit.

When to Decide to Swarm, Pair, and/or Split

- While a lot of time can be spent on pair programming, it does make sense to split the work when folks understand how the work will be carried out, and the work to be done is largely prescriptive.
- Once the story has been jointly tasked out by both engineers, the engineers may choose to tackle some tasks separately and then combine the work together at the end.
- Pair programming is more helpful when the engineers do not have perfect clarity about what is needed to be done or how it can be done.
- Swarming is done when the two engineers assigned to the story need an additional sounding board or need expertise that other team members could provide.

Benefits of Increased Collaboration

Knowledge sharing and bringing ISE and customer engineers together in a 'code-with' manner is an important aspect of ISE engagements. This grows both our customers' and our ISE team's capability to build on Azure. We are responsible for demonstrating engineering fundamentals and leaving the customer in a better place after we disengage. This can only happen if we collaborate and engage together as a team. In addition to improved software quality, this also adds a beneficial social aspect to the engagements.

Resources

- How to add a pairing custom field in Azure DevOps User Stories - adding a custom field of type *Identity* in Azure DevOps for pairing

- On Pair Programming - Martin Fowler
- Pair Programming hands-on lessons - these can be used (and adapted) to support bringing pair programming into your team (MS internal or including customers)
- Effortless Pair Programming with GitHub Codespaces and VSCode

---

Last update: August 22, 2024