For this discussion, we will explore best practices and tools for testing improvement and continuous integration.

In your initial post, address the following:

* What is your familiarity with continuous integration tools? Would you say you are highly versed with these tools, have some experience, or do not have any experience? (It is okay if you do not have any experience; that is why we are discussing them here.)
* If you do have experience, what is a continuous integration tool that you have used, and what is the context in which you used it? What are the pros and cons of it?
* If you do not have experience, what is a continuous integration tool you would like to learn more about (e.g., CircleCI, Jenkins)?

In your responses to at least two peers, compare and contrast your experience with continuous integration tools. Perhaps you have used a tool, but in a different context, and would like to share your experience. Provide resources and/or additional ways to use a continuous integration tool, if possible.

I do not have any experience using continuous integration tools. Before this chapter, I wasn’t even aware of “CI” tools. Still, reading about it has piqued my interest to learn more about it. Using tools to streamline development processes & contributing to team collaborations seems to be the primary focus.

Some light research states that Jenkins is known as a widely used open-source CI tool that users have praised for its flexibility & extensive plugin support. Being open source, I would assume that small teams love the cost saving aspect of it. There also seems to be a bit of an active community to troubleshoot and support, which helps prolong the use of life for the tools. I looked at some beginner guides, curious to set up, and it seemed a bit daunting. Not something I could easily set up in a 20-minute session.

Our prompt mentions CircleCI, again another tool to look up and read more about. It’s also known to be simplistic, but it is more user friendly & cloud based. Cloud based integration allows it to be ran over widespread range of units with different specs. I’d be interested to learn more about its day-to-day usage and how it varies from a team using Jenkins.

*CI/CD: Complete Guide to Continuous Integration and delivery*. Codefresh. (2024, December 11). https://codefresh.io/learn/ci-cd/

Jain, P. (2020, May 13). *An introduction to circleci*. Medium. https://medium.com/xebia-engineering/an-introduction-to-circleci-aa9464a86673

Hey there Alyssa, second week, so far, so good I’d say.

I share your sentiments about the lack of experience using continuous integration tools. Sometimes I feel like new tools, & concepts are being presented at breakneck speeds due to how our school’s semester are set up. Sometimes, I feel the information is not fast enough either. I think you nailed it on the head through your explanation of how it can be used to help understand different interfaces & components.

The interest in learning the unknown is shared amongst us as students. It’s simply not enough to use a program in today’s days & age, without trying to push its limits, view some source code, and see how it fits into the bigger picture of things. I don’t have an y personal experience to share, but I’m hoping to at least be understanding of what I see placed in front of me in the future.

Hey there Max, I really enjoyed reading your post!

First things first, my lack of experience is on full display here, as I could only recognize about 2 terms from what you posted, the rest were interesting to look up and read about them. It seems you are already well established in the IT field and showcase some valuable experience. Your description of the rolling deployments sounds somewhat similar to a task I preferred back at a summer job, where we went through different parts of the college, pushing out updates in series to computers. A nightmare that should have been easy, so many had received corrupt images, and needed to be manually wiped and installed. So yeah, something going wrong during deployment indeed.