Version Control Guidance

I found three sources which discuss version control when it concerns DevOps. One source was from Microsoft, one from Atlassian, and one from Medium written by Smit Gabani. I made sure the sources discussed version control in DevOps rather than just version control in documents as it is a more specific subject.

The Microsoft article discusses the importance of creating workflows, working with versions, synchronizing versions, and automating features. The article stresses the importance of keeping everything synchronized and up to date so that any updates may be built upon any previous code without causing confusion. Automation is mentioned as a feature that can help save time and create consistency when it comes to tasks like code analysis. In my opinion this isn’t an absolute necessity for version control but is certainly nice to have if a team has the budget and time to set it up. Otherwise, these tasks can be done manually on a schedule.

Atlassian stresses the importance of protecting the source code, explaining that version control helps prevent the source code from being lost due to various circumstances. An interesting note from this article not mentioned in Microsoft’s is that version control should never impede another developer’s work by protecting another’s. It should aim to preserve changes while allowing for further changes to be made. While Atlassian does not specify certain steps or guidance for version control, it does suggest that using a version control software would be best practice for implementing version control within a fast-paced environment or one that has many developers involved.

Smit Gabani’s article from Medium discusses the nuances of using version control when working individually as well as within a team, which is an interesting aspect not covered in the other articles. This article also discusses the negative possible side effects of not utilizing version control in different scenarios, such as not being able to accurately trace down the origins of a bug within the code. Situations like this can cause a lot of wasted time and energy down the road without version control. While this article doesn’t give guidance for version control steps explicitly, it does explain the differences between centralized and distributed version control systems and when they would best be applicable.

While it was difficult to find sources that explicitly suggested steps for version control, I feel that these articles all mentioned different important aspects of implementing version control that should be considered. I believe the most important aspects covered were preserving the source code and determining which type of version control (centralized or distributed) a team should use. These concepts are important because they are the essence of version control and the method in which it is accomplished can determine its success within the environment.

Sources

*What is version control? - azure DevOps*. Azure DevOps | Microsoft Learn. (2022, November 28). https://learn.microsoft.com/en-us/devops/develop/git/what-is-version-control

*What is version control: Atlassian Git Tutorial*. Atlassian. (n.d.). https://www.atlassian.com/git/tutorials/what-is-version-control

Gabani, S. (2024, February 3). *Version control, source control, platforms and their role in DevOps*. Medium. https://medium.com/@smitgabani/version-control-source-control-platforms-and-their-role-in-devops-3d115e06a10d