Ryan Norrbom

CSD380

Assignment 3.2

August 29th, 2024

Version Control

Version control is an essential process for managing codebases while supporting team collaboration and maintaining the integrity of software development projects. As version control and the many solutions and methodologies that rely on it grow in adoption, ever-changing guidelines exist to help teams navigate the complexities of version control systems (VCS) like Git. Below, we'll compare guidelines from three resources: Atlassian, University of Washington, and Harvard.

When working on a project and using version control, each resource highlighted that it is important to keep the following guidelines in mind. Commit frequently by making small, frequent commits to keep changes manageable and easier to debug. Use meaningful commit messages that explain the change's purpose to improve other developers' ability to understand the code's functions. In addition, developers who use version control should leverage branching for features or bug fixes and merge them into the main branch only after testing. Furthermore, documentation of changes in commit messages and external documentation will help maintain a comprehensive project record. Finally, it is best to tag versions of the code to identify milestones, releases, or important project states.

There was another similarity to each resource that stood out above all else. Each resource discussed how vital it is to use version control to enable teams to collaborate better, merge their ideas, and iterate their code. When selecting a version control tool and methodology, it is best to consider the current and desired structure of the team. The differences between the resources were minute but would stand out depending on the team.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Harvard** | **University of Washington** | **Atlassian'** |
| **Approach to Experimentation** | Branch for testing new features away from main code. | Strict branching procedure to avoid conflicts. | Use submodules for isolated experiments. |
| **Integration Strategy** | integrate changes only after thorough testing to maintain stability. | Continual integration and frequent merges. | Focuses on modular integration to manage dependencies. |
| **Security and Privacy** | Store sensitive data in secure environments with version control. | Review access permissions regularly. | Use built-in security features of the tools |
| **Collaboration Model** | Promotes a collaborative model with clear documentation and roles. | Set clear rules for pull requests and code reviews. | Use tools like Jira |
| **Adaptability** | Remain adaptability to different project types and sizes. | Standardize workflows workflow to minimize errors. | Flexible and Custom workflow. |

As CI/CD systems become more advanced, integrated, and available, the role of automated testing and quality checks has expanded. In turn, it reduces the need for some of the manual interventions recommended today. This emphasis on manual documentation of changes outside of commit messages may seem redundant as more software projects align with these more modernized project management and tracking tools, such as Jira, which automatically link commits, issues, and documentation. Furthermore, Artificial Intelligence systems can assist developers in documenting their code and highlighting changes during the CI/CD process, creating a consistent operation around code and commit notes.

The guidelines outlined by each resource are helpful as they address software development needs: safeguarding code, collaboration, automated and manual quality checks, and upkeeping clarity. They align, with variation, to the best practice models that support agile and DevOps methodologies, where rapid iteration and continuous improvement support increasingly complex solution-making. Adapting these guidelines to your desired workflow will improve the efficiency and reliability of the version control process and help teams maintain a clear, organized, and manageable codebase. Finally, adopting these practices ensures that version control is a foundation for development, scaling effectively with project demands and team growth.

GitHub URL: <https://github.com/GeraltOfCodea/csd-380>

**References**:

Atlassian. (n.d.). *Submodules: Core concept, workflows and tips: Atlassian Git Tutorial*. <https://www.atlassian.com/git/articles/core-concept-workflows-and-tips>

Version control concepts and best practices. (n.d.). <https://homes.cs.washington.edu/~mernst/advice/version-control.html>

*Version control*. Data Management. (n.d.). <https://datamanagement.hms.harvard.edu/collect-analyze/version-control>

Kim, G., Debois, P., Willis, J., Humble, J., Forsgren, N., & Allspaw, J. (2021). *The devops handbook: How to create world-class agility, reliability, & Security in Technology Organizations*. IT Revolution Press, LLC.