Kevin Ramirez

CSD 380

Prof. Woods

March 5 2025

Version Control Guidelines

Version control is crucial for managing code in teams and software companies; it allows for smooth collaboration and keeps a tab on progress. Many different organizations have their own approach when it comes to version control and how to implement it effectively. In this paper, I will go over three sources, their recommendations, and the most important practices to follow.

Key Sources and Their Recommendations

1. Git's Official Guide (2023)

This guide highlights the following best practices:

- **Make Small, Frequent Commits** – Breaking changes into smaller chunks makes them easier to review and revert if needed.
- **Write Clear Commit Messages** – Writing clear commit messages helps all other developers identify your purpose.
- **Use Branches for New Features or Fixes** – Keep any new updates separate from the original code.
- **Always Pull Before Pushing** – Syncing with the latest changes reduces merge issues.

2. Atlassian's Git Tips (2022)

Here are Atlassian's tips:

- **Follow Branch Naming Conventions** – It's important to keep the naming convention readable.
- **Prefer Rebasing Over Merging** – Keeps the commit history cleaner and easier to follow.
- **Require Code Reviews Before Merging** – Ensure senior engineers look over your code to reduce errors.
- **Avoid Storing Large Files in Git** – Use Git LFS for big binary files.

3. Microsoft's Azure DevOps Advice (2021)

Microsoft's guide focuses on security and automation:

- **Lock the Main Branch** – Require pull requests and approvals to prevent direct pushes.
- **Tag Important Releases** – Use version numbers to mark releases.
- **Automate Testing with CI/CD** – Run tests automatically before allowing merges.
- **Never Force Push to Shared Branches** – Rewriting history can cause problems for teammates.

Comparing Guidelines and Identifying Outdated Practices

- **Rebase vs. Merge** – Atlassian suggests rebasing for a cleaner history, while Microsoft prefers merge commits for better traceability.
- **Handling Large Files** – Atlassian explicitly warns against storing binaries in Git, while Git's docs leave it to the developer's discretion.

My Top Version Control Guidelines

After reviewing these sources, here’s my list of the most important practices:

1. **Commit Small Changes Often** – Makes tracking and reverting easier.
2. **Use Descriptive Commit Messages** – Helps others understand why a change was made.
3. **Work in Feature Branches** – Keeps the main branch stable and reduces conflicts.
4. **Require Pull Requests & Reviews** – Ensures code quality before merging.
5. **Protect the Main Branch** – Prevents accidental pushes and enforces workflow rules.
6. **Rebase Instead of Merge (When Safe)** – Keeps history cleaner without unnecessary merge commits.
7. **Tag Releases Clearly** – Makes it simple to find stable versions.
8. **Never Force Push Shared Branches** – Avoids disrupting others’ work.

Sources

1. Ernst, M. D. (n.d.). *Version control best practices*. University of Washington. Retrieved April 5, 2025, from https://homes.cs.washington.edu/~mernst/advice/version-control.html

2. Perforce Software. (n.d.). *8 version control best practices*. Perforce. Retrieved April 5, 2025, from https://www.perforce.com/blog/vcs/8-version-control-best-practices

3. GitLab. (n.d.). *Version control best practices*. GitLab. Retrieved April 5, 2025, from https://about.gitlab.com/topics/version-control/version-control-best-practices/