Scott Cacal

1.26.2025

CSD 380

Assignment 3.2

Version Control Guidelines

Version control (VC) guidelines has shifted over time. In Concurrent Version Systems (CVS) where code would be stored in a central server the shift is being made towards a distributed model where developers work on local instances of the repository – systems such as Git use this model and it’s popularity is driving the need for advancements in version control. Nevertheless it should be noted that the importance of maintaining code integrity and stability continues to be a priority and in this report we will discuss articles promoting relevant version control guidelines (as late as mid 2024) that should be practiced and we will form our own list from this analysis.

Below is a comparison table of the three articles from; daily.dev, Filestage, and Helpjuice who’s artcles prove useful insight as what should be prioritize when establishing or implementing a version control system and guidelines to maintain.

|  |  |  |
| --- | --- | --- |
| daily.dev | FILESTAGE | Helpjuice |
| Set a VC plan (what system) | Versioning Policies (Naming conventions consistent) | Versioning Policies |
| Versioning Policies | Review frequently – with boundaries | Train or maintain knowledge on system |
| Diff. branches for diff. features | Centralize feedback to guarantee alignment | Review Freequently |
| Review frequently | System for reviewers should be easy | Integrate VC into workflow |
| Ensure document aligns code |  | Integrate Automation |
| Integrate Automation |  |  |
| Prioritize Security Measures |  |  |
| Back up frequently and test |  |  |

Note that green represents they align across all three articles and the yellow represents alignment in just two.

What was wonderful about the article by daily.dev is they provided such an extensive list to build from with broad points that we can relate to the other two, Filestage and Helpjuice, articles. While I typically prefer a simplified approach to rules or guidelines my list has become slightly longer than Filestage and Helpjuice.

Below are the guidelines I would practice:

* **Versioning Policies** such as naming convention, format in commiting code, and establishing what is the latest version.
* **Reviewing Frequently** to ensure versioning policies are maintained as well as the establshing the latest version of code.

These two points were consistent across the three articles and circling back to the beginning of this report, I can recognize this as foundational across all methods of version control.

* **Integrate Automation** as additional tools will be available and automation of backups or checkin notices can prevent any redundant work.

The remainting three points come from each articles.

* **Prioritzing Security Measures** may already be practiced by the organization but enforcing it around localized source code must take priority.
* **A simplified and centralized feedback system for reviewers** so concepts, ideas, or reviews are not bounced around between few
* **Integrate VC in workflow** from automations to personal checks as a developer will further increase effacacy and efficiency of the system.

To drive the point of integration of version control into ones workflow and to wrap this all up. Initial exposure to any VC system can be daunting, but with guidelines such as the ones discussed, frankly, I believe practice and exposure are the best ways to establish the confidence to work with them.

References:

“Documentation Version Contro: Best Practices 2024” . 2024 . https://daily.dev/blog/documentation-version-control-best-practices-2024 . daily.dev

Lott, Ethan . “Document version control best practices and examples” . 2025 . https://filestage.io/blog/document-version-control/ . FILESTAGE

Mugwe, Rush . “A Guide to Document Version Control” . 2024 . https://helpjuice.com/blog/document-version-control#:~:text=Version%20control%20is%20crucial%20for%20a%20knowledge,are%20updated%2C%20and%20new%20information%20is%20discovered. . Helpjuice