

Version Control Strategy Proposal

1. Introduction

This document outlines a recommended version control strategy for 100AND10%. We currently lack a standardised version control platform, which creates potential risks in terms of code management, which Jack and I have already experienced working together on UKactive.

As the development team grows, implementing a structured version control system is essential for maintaining code quality, and ensuring reliable workflows. In addition it makes tackling larger projects much easier.

This document evaluates two primary options:

Self-hosted Git infrastructure

GitHub Pro (cloud-hosted Git platform)

Following an evaluation of cost, operational overhead, security considerations, and integration capabilities, **GitHub Pro** is recommended as the preferred solution.

2. Objectives of Implementing Version Control

The adoption of a formal version control system would provide several key benefits:

- Centralised management of all project codebases
- Improved collaboration between developers
- Version history and rollback capability
- Secure backup of source code
- Support for modern development workflows
- Integration with deployment and hosting platforms

For a distributed team, these capabilities are particularly important for ensuring consistent development practices, this ultimately speeds up work as our repositories become our single “Source of truth” for current and future projects. This has the added benefit increasing the quality of our output as well our general productivity.

3. Options Considered

3.1 Self-Hosted Git

Self hosting a git environment is relatively simple, I have implemented both Gitea and Gitlab in the past both a reliable options, with Gitlab being more feature complete.

This approach would give the organisation full control over the infrastructure and repository hosting. However, this approach introduces several operational responsibilities, including:

- server provisioning and maintenance
- security patching and vulnerability management
- backup and disaster recovery procedure
- authentication and access control configuration
- ongoing monitoring and infrastructure management

This is not a hard block to self-hosting, and by utilizing cloud services the hardware is managed for us, but ultimately we are doing a lot of work while still sacrificing some control.

3.2 GitHub Pro

GitHub Pro provides a fully managed Git hosting platform with extensive collaboration and security features.

Key capabilities include:

- private repositories
- pull request workflows
- branch protection rules
- issue tracking and project boards
- code review tools
- integrated CI/CD support via GitHub Actions
- built-in security scanning

Team

Advanced collaboration for
individuals and organizations

\$ **4** USD per user/month

Continue with Team

← Everything included in Free, plus...

- > Access to GitHub Codespaces
- > Repository rules
- > Multiple reviewers in pull requests
- > Draft pull requests
- > Code owners
- > Required reviewers
- > Pages and Wikis
- > Environment deployment branches and secrets
- > 3,000 CI/CD minutes/month
Free for public repositories
- > 2GB of Packages storage
Free for public repositories
- > Web-based support

FEATURED ADD-ONS

- > GitHub Secret Protection
- > GitHub Code Security

GitHub also benefits from a mature ecosystem and widespread adoption across the software industry.

4. Security Considerations

One of the primary considerations when evaluating cloud-hosted solutions is the potential increase in organisational attack surface. However, for a fully remote team such as 100AND10%, this risk is minimal. Developers are already accessing systems via the public internet, and GitHub provides enterprise-grade infrastructure security that would be difficult to replicate internally.

By using GitHub's managed platform the infrastructure and platform security concerns are largely handled by GitHub themselves. This includes things such as redundancy, DDoS protection, and vulnerability management for the underlying service.

If we were to self-host a Git server we would instead be responsible for maintaining the infrastructure, managing security updates, implementing backup strategies, and monitoring the service ourselves. For a development team of our current size this would introduce operational overhead without delivering much practical benefit.

GitHub also provides a number of security features out of the box such as two-factor authentication, role-based access controls, audit logs, and secret scanning. These features help reduce the risk of credential exposure and unauthorised access to repositories.

5. Conclusion

For a small development team consisting of two developers, the cost difference between GitHub Pro and a self-hosted solution is significant. With self-hosted options costing from £15+ per month not including usage and other costs, Github Pro costs £4 per month per user for a decent suite of features for two developers.

After evaluating both self-hosted and cloud-hosted approaches, GitHub Pro provides the most practical and cost-effective solution for the organisation's current team size and distributed working model at present. We will inevitably outgrow it at which point we can look at the enterprise tier or self-hosting. However as standardised version control isn't yet part of our working practices, it makes sense to take a more hand-held option to start with.