



Software for version management is used to record the change history of files and directories and, if necessary, to be able to reset them to a previous work status. Often, additional tools are offered to split work steps on the same file and to bring different versions of this file together again. This means that several people can collaborate on a project without accidentally deleting important data. In the following, we will discuss what the advantages of the Git system are and why *GitLab* stands out in comparison to other software solutions in order to organize and version source code.

### Version management software using the example of Git

Git is a distributed version control system that records the change processes of files. It impresses with its decentralized work on projects in which an official repository with the shared version of the files exists on a server and a copy can be used on many other local computers. No constant internet connection is required for this. With the local working copies, branches can be created from a project and, if necessary, merged with the official repository again. This offers many advantages in collaborative work in projects and is considered to be particularly quick and efficient when changing only small text-based files.



As an alternative to Git, there are many other systems for versioning files and directories. SVN (*Apache Subversion*) should be mentioned, which is also widespread. It is considered an easier-to-use central system and is better able to handle large and complex files. In practice, source code and documents are therefore often managed using Git, while images, videos and other large binary formats that cannot be read with a text editor are stored on an SVN system.

### GitLab – A web application for managing software projects

*GitLab* is a web application for version control based on Git. The open-source software has the advantage that it can be set up as a system free of charge on your server. Projects can be organized by yourself and access to the system can be configured. Many automated functions help in the software development process, from planning, creation, and build, up to verify, security testing, deploying, and monitoring. Alternatively, there are also related software solutions such as *Gogs* or *Gitolite*, which also allow you to set up locally hosted repositories, but which do not offer the same functionality as GitLab.



There are also well-known public platforms such as *GitHub* and *Bitbucket*, which also host software projects and use Git as a basis. Even if they make a large contribution to the *open-source* movement, they have two major disadvantages: 1) The software projects usually have to be made public if you do not pay a certain surcharge, and 2) the corresponding data is on external servers. Since sensitive data is often dealt with during the project phase, such as copyrighted images or videos or personal results data from a survey, these must not simply be made public or stored on external storage devices. *GitLab* allows local editing of files but also prepares them so that the transfer to *GitHub*, for example, is not a problem at the end of the project. This is supported by the fact that *GitLab* and *GitHub* have a comparable infrastructure. The lowest level to be managed is represented by the repositories, i.e. the individual projects. These can be organized in an indefinite number of groups and sub-groups. This has the advantage that not only a small workgroup can use *GitLab* to structure their projects in a few sub-areas but also a large institution can separate projects at multiple organizational levels.

# Best Practice: Organisation und Versionierung von Quellcode



## The shift from CVS to Git - in conversation with the Webis organization

The Webis Group (short for web technologies and information systems; website: <https://webis.de>) is a research group based at the Bauhaus University Weimar. Their research contributes to web mining and retrieval, machine learning, computational linguistics, and symbolic AI. The software CVS (*Concurrent Versions System*) was used here for several years to organize the own software projects and files with version control. In 2015 it was decided to transfer the software projects to Git and make them more accessible with the *GitLab* web interface.



Digital Bauhaus Lab, Photo: Tina Meinhardt

One of the reasons for the change was the fact that CVS is currently no longer being developed and certain mechanisms for software development are unfavorable here. For example, there may be inconsistencies when updating a project because CVS only records changes at the file level, while Git updates the entire project at the commit level. The procedure is thus closer to the actual software development procedure, in which a commit can correspond to a release status of the software. Another problem is, for example, the lack of tracking of name changes, which can lead to a split in the history of a project. Git uses heuristic rules to compare content and detect file name changes.

The actual *GitLab* web application has other diverse advantages. It gives you an easy overview of the software projects and the access can be determined depending on the user group. Features such as the rendering of the *README* files and the creation of wiki pages allow the projects to be well documented and problems can be reported, tracked and dealt with using an issue tracker.

Webis also builds on the configurability of the interfaces from Gitlab by using the local authentication interface LDAP (*Lightweight Directory Access Protocol*) for the login. This allows both employees and students to log into the system with their known university login details without having to come up with and remember new login details.

## Open-source and community

*GitLab* is open-source software that was developed with the programming languages Ruby and Go. It is therefore freely available and can be configured completely according to your own needs, but if necessary, also expanded or changed if you have the appropriate programming skills. If you notice a bug in the software, you can report it via a forum or a bug tracker. Thanks to the developer *GitLab Inc.* and a large community, you can usually rely on a quick solution.

There are several ways to use *GitLab*. The primary and previously presented variant is to set up the software on its own server. Since this is associated with an increased technical effort and the software must first be set up and configured securely, it is advisable to ask the data center or the nearest IT support for this matter. Instances of *GitLab* often already exist at central facilities, so creating a new instance is not a major problem. Furthermore, *GitLab Inc.* has also been offering SAAS (*Software as a Service*) since 2012 to host private and public repositories. Support can be booked for this purpose.

If you have any questions about *GitLab* or other software repositories, please contact Mr. Kevin Lang, Research Data Management Contact Point at the Bauhaus University Weimar ([kevin.lang@uni-weimar.de](mailto:kevin.lang@uni-weimar.de)).



Do you have any questions about this Best Practice or would you like to suggest another one?

Please contact us: [info@forschungsdaten-thueringen.de](mailto:info@forschungsdaten-thueringen.de)