УДК 004.4’2

Мороз А. – студент
Центральноукраїнський національний технічний університет

ОБҐРУНТУВАННЯ ВИБОРУ РОЗПОДІЛЕНОЇ СИСТЕМИ КОНТРОЛЮ ВЕРСІЙ ДЛЯ РЕАЛІЗАЦІЇ ІТ-ПРОЄКТІВ

Науковий керівник: канд. техн. наук Доренський О. П.

Moroz A. – Student
Central Ukrainian National Technical University

CHOICE OF DISTRIBUTED VERSION CONTROL SYSTEM FOR IT PROJECTS REALIZATION JUSTIFICATION

Supervisor: PhD in Information Technology Oleksandr Dorenskyi

Ключові слова: система контролю версій, Mercurial, Darcs, Fossil, GitHub, GitLab.
Keywords: Version Control System, Mercurial, Darcs, Fossil, GitHub, GitLab.

Version control systems (VCS) is usually used for documenting, tracking and control the projects, which several people are working on at once. Distributed VCS save all the files on the local repository and synchronize with similar local repositories on other computers. The local repository is stored in the working directory [1]. Nowadays, there are many realizations of VCS, which are successfully used by appointment. Therefore a problem of justified choice of the version control system for IT-projects springs up.

The distributed VCS Mercurial, Darcs, Fossil, GitHub and GitLab are analyzed in work. The first system has high productivity of work with the repository, data is stored compactly and indexed, HTTP and SSH are used, information is transmitted compactly. The number of developers who takes part in the project is not limited. The SHA1 algorithm is used, data does not replace but adds. Also, there is a fast algorithm of checking repository integrity, a web-server is integrated, which allows having access to the repository through web-interface. This VCS has remedies which simplify migration from other version control systems, supports several repository organization models(centralized CVS-similar, decentralized hierarchical and distributed semi-hierarchical), external handlers and additions [1].

The main element of Darcs is not a branch, but a patch. Changes are stored locally. Changes can be cancelled if the repository was not published or others do not have access to it. This VCS supports test suite integration; it is possible to check the program for bugs before publishing changes, use any server for access to the repository. Darcs does not write access to files (you need to do it by yourself), the possibility to move and rename files, without losing their stories is realized [2].

All the functionality is realized in one performed file, the size of which is less than one megabyte. This VCS includes a mistakes tracking system, editor and wiki-pages storage for more effective and comfy work with the project, a simple HTTP-server, with help of which the main work with the repository is performing, is realized. Fossil web-interface can be flexibly set by a user, built-in HTTP-server supports protocol of work with external apps CGI, in accordance, the functionality of granted interface can be much expanded. In this VCS all repositories are automatically checked for integrity and conflict absences, repository reliability is supported with using DBMS SQLite3 [3-4].

In GitHub, it is possible to create a new branch, merge it and then delete it for every new feature. Also using several branches(the main, the test, for daily work). This VCS is fast(all operations performs locally), many users can work with one repository at once. Besides GitHub has an interim zone for files and a license with open code [5-6]. GitLab allows tracking problems, supports integrations, provides an easy label enabling, has corporative settings, provides a capability to cancel the commit. Access is provided by people roles. This CVS creates confident issues, which are seemed to participants with certain access level, represent monitoring panel for time analyze, planning and monitoring, has monthly updates. GitLab is more oriented for integrating as many functions into a platform for DevOps process as possible [5].

Results of CVS analyse by characteristics of speed, repository integrity checking, mistakes tracking, confident issues are represented in the table 1 [4, 8].

Table 1

	Mercurial	Darcs	Fossil	GitHub	GitLab
Speed	+	–	–	+	+
Checking repository integrity	+	+	+	+	+
Mistakes tracking	+	+	+	+	+
Confident issues	+	–	–	–	+

Summarizing received analyze result we can highlight the advantages and disadvantages of CVS using. To advantages we can include: 1) every developer works with own repository; 2) decision about branch merging can be taken by the head of the project; 3) no need in internet connection. The disadvantages are: 1) impossible to control file access; 2) general enumeration of file version is absent; 3) much more server disk storage is needed for saving; 4) ability to block files is absent. Distributed VCS allows people together to work with one file and to create some parts of the project; members of a big project can work on own parts (even if they don't have a stable internet connection). At the same time, VCS have disadvantages, because of which it is worth to think about using distributed VCS. Storing full version history of the project at every computer, disability of blocking file or group of files are among them.

From the results of the work, it follows, that during project realization, Mercurial or GitLab should be chosen for using because of characteristics such as speed, repository integrity checking, mistakes tracking and confident issues.

References

1. Mercurial wiki. Web-site. URL: www.mercurial-scm.org (Last accesed: 16.03.2020).
2. Darcs. Web-site. URL: https://uk.wikipedia.org/wiki/Darcs (Last accesed: 16.03.2020).
3. Version control system: Fossil, part I. Web-site. URL: https://habr.com/ru/post/235369/ (Last accesed: 17.03.2020).
4. What is Fossil? Web-site. URL: http://www.fossil-scm.org/index.html/doc/trunk/www/index.wiki (Last accesed: 17.03.2020).
5. GitLab versus GitHub. Web-site. URL: https://senior.ua/articles/gitlab-protiv-github.
6. Git About. Web-site. URL: https://git-scm.com/about (Last accesed: 18.03.2020).
7. GitLab Enterprise Edition. URL: https://gitlab.com/help (Last accesed: 18.03.2020).
8. Feature Highlight: Confidential issues. Web-site. URL: https://about.gitlab.com/blog/2016/03/31/feature-highlihght-confidential-issues/ (Last accesed: 04.04.2020).