Paper1

Title: Version Control System: A Review

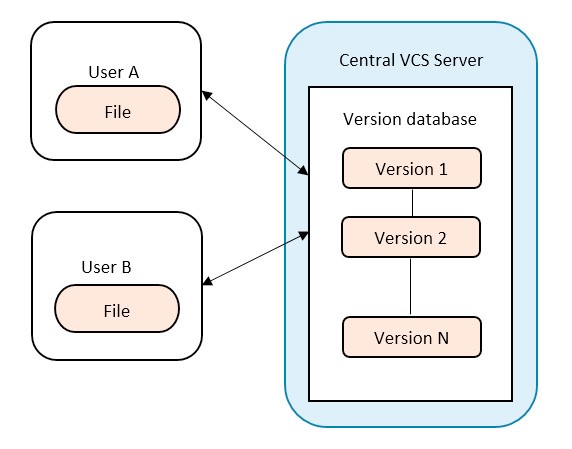
Authors: Nazatul Nurlisa Zolkifli, Amir Ngah\*, Aziz Deraman

Published in :

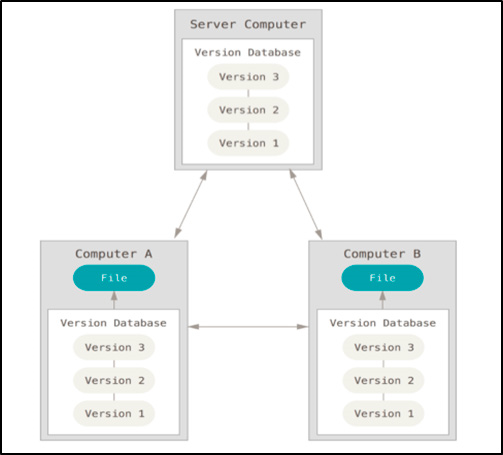
Introduction to VCS:

* Version Control System (VCS) is a system that manages the development of an evolving object.
* VCS is also known as Revision Control System

Centralised VCS:



Distributed VCS:



Comparison between Centralised and Distributed

Table 1. Comparison between CVCS and DVCS.

|  |  |  |
| --- | --- | --- |
| Version Control System | CVCS | DVCS |
| Repository | There is only one central repository which is the server. | Every user has a complete repository which is called local repository on their local computer. |
| Repository Access | Every user who needs to access the repository must be connected via network. | DVCS allows every user to work completely offline. But user need a network to share their repositories with other users. |
| Example of VCS Tools | Subversion, Perforce Revision  Control System | Git, Mercurial, Bazaar, BitKeeper |
| Software Characteristics that suitable | 1. Projects that allow only several users to contribute to the software development. 2. Team located in a single site. | 1. DVCS is suitable for a single or more developers because the project repository is distributed to all the developers and this ability offer a great improvement for the projects. 2. It also can be applied for a small or big software projects because it makes less difficult for normal users to contribute to the development. 3. Team located in multiple site or different countries and different timezones. |

Applications:

* Software merging
* Collaboration Modelling
* Software changes
* Software branching
* Open Source Software Projects

Conclusion:

Software developers should have a rudimental understanding of what VCS is and which type of VCS suits them. The adoption of a VCS is a must in software development. It helps software developers manage their codes easily because it is common to have a lot of changes involving addition or deletion of features. In order to adopt a VCS, a software developer must know and perfectly understand which approach should be used as it will affect the whole project and team. It is also important for them to have the knowledge of different approaches of VCS because the various approaches will affect their software development process differently.

Paper:2

Title:How Do Centralized and Distributed Version Control Systems Impact Software Changes?

Authors: Caius Brindescu, Mihai Codoban, Sergii Shmarkatiuk, Danny Dig

**Table 1: Demographics of survey respondents**

1. **Programming experience ( years )**

*<* 2 2 - 5 5 - 10 10 - 15 15 - 20 *>* 20

1.83% 11.10% 30.49% 30.61% 13.90% 12.07 %

1. **Project type**

|  |  |  |  |
| --- | --- | --- | --- |
| Proprietary Open source Research software software project | | Personal project | Other |
| 85.09% 6.97% 4.64% | | 3.06% | 0.24 % |
|  | **(c) Team size** |  |  |
| 1 2 - 5 | 6 - 10 11 - 25 | 26 - 100 | *>*100 |
| 5.87% 42.30% | 23.72% 15.65% | 8.19% | 4.28% |

**d**

**(**

**)**

**Project**

**age**

*<*

6

mo

6

mo

-

1

yr

1

yr

-

2

yrs

*>*

2

yrs

%

21.27

%

46.82

%

13.33

%

18.58

**VCS**

**used**

**predominantly**

**(**

**e**

**)**

Microsoft

Git SVN Hg CVS Other TFS

52.68% 20.37% 12.07% 8.54% 1.10% 5.24 %

**Table 2: Repository corpus.**

|  |  |  |  |
| --- | --- | --- | --- |
| Type. Repositories | Commits | Authors | Total  LOC  changed |
| SVN 52 | 95571 | 451 | 270 M |
| Hybrid 29 | 151004 | 2249 | 89 M |
| Git 51 | 111725 | 3190 | 50 M |
| Total 132 | 358300 | 5890 | 409M |

# RESULTS

**Observation 1:** DVCS repositories have a smaller commit size than CVCS repositories, in terms of lines of

**Observation 2:** 76% of the developers split their com

mits. The percentage is higher for distributed version control systems (81.25%), compared to centralized ones (67.89%).

**Observation 3:** Overall, developers choose to split their commits using the intent of change.

**Observation 11:** Most projects use an Issue Tracking System

**Observation 17:** Commit size tends to become smaller as projects get older

**Observation 13:** Large teams squash commits more often

# CONCLUSIONS

In this paper we present the first in-depth study to measure the impact of DVCS on software change. To this end we ran a survey with 820 participants and analyzed a corpus of 132 repositories.

We found that the use of CVCS and DVCS have observable effects on developers, teams and processes. The most surprising findings are that (i) the size of commits in DVCS was smaller than in CVCS, (ii) developers split commits (group changes by intent) more often in DVCS, and ( iii ) DVCS commits are more likely to reference issue tracking labels. These show that DVCS contain higher quality commits compared to CVCS due to their smaller size, cohesive changes and the presence of issue tracking labels.