

Version Control System

Reece Donovan

B.Sc. Computer Science - Final Year Project

Prof. Ken Brown

University College Cork

October 22, 2022

Abstract

Declaration of Originality

Acknowledgements

Contents

1	Introduction	2
2	Background	3
2.1	What does a VCS look like?	3
2.1.1	What does a VCS aim to do?	3
2.1.2	Types of VCS	3
2.2	Existing Version Control Systems	3
2.2.1	Concurrent Versions System (CVS)	3
2.2.2	Subversion (SVN)	4
2.2.3	Perforce	4
2.2.4	Git	4
2.2.5	Mercurial	4
2.3	Summary of Key Features	5
2.3.1	Repositories	5
2.3.2	Commits	5
2.3.3	Branching and Merging	5
2.3.4	Pulling and Pushing	5
3	Design	6
4	Implementation	7
5	Evaluation	8
6	Conclusion	9

Chapter 1

Introduction

Chapter 2

Background

2.1 What does a VCS look like?

A Version Control System (VCS) provides

2.1.1 What does a VCS aim to do?

A VCS aims to provide a

2.1.2 Types of VCS

There are two main types of VCS: centralized and distributed.

Centralized Version Control Systems

Distributed Version Control Systems

2.2 Existing Version Control Systems

2.2.1 Concurrent Versions System (CVS)

Concurrent Versions System (CVS) was developed in the 1980s by Larry Wall and was the first VCS to become widely used for collaborative software development. Concurrent Versions System is a centralized VCS, meaning that there is a single server that stores the entire history of the project. The server is the only place where files can be added, removed or modified.

Advantages

Disadvantages

2.2.2 Subversion (SVN)

Subversion (SVN) is a centralized VCS that was developed by CollabNet in the year 2000. Subversion was developed to replace CVS, which was becoming increasingly difficult to maintain.

Advantages

Disadvantages

2.2.3 Perforce

Perforce is a centralized VCS that was developed by Perforce Software in 1995. Perforce is a commercial VCS that is used by many large companies, such as Google, Adobe and IBM. It is still one of the largest version control systems in use today.

Advantages

Disadvantages

2.2.4 Git

Git is a distributed VCS that was developed by Linus Torvalds in 2005. Git is a free and open source VCS that is used by many large companies, such as Google, Facebook and Twitter. It is one of the most popular version control systems in use today.

Advantages

Disadvantages

2.2.5 Mercurial

Mercurial is a distributed VCS that was developed by Matt Mackall in 2005. Mercurial was developed with the same goal as Git, to maintain the Linux kernel project.

Advantages

Disadvantages

2.3 Summary of Key Features

2.3.1 Repositories

2.3.2 Commits

2.3.3 Branching and Merging

2.3.4 Pulling and Pushing

Chapter 3

Design

Chapter 4

Implementation

Chapter 5

Evaluation

Chapter 6

Conclusion