OOP Report

Repositories

Submitted to:

Sir Khalid Rasheed

Farhan Zafar Khan

1445108

# Contents

Table of Contents

[Contents 2](#_Toc429645507)

[1.0 Introduction 1](#_Toc429645508)

[2.0 Repositories 1](#_Toc429645509)

[2.1 Definition 1](#_Toc429645510)

[2.2 Purpose of Repositories 1](#_Toc429645511)

[3.0 Version Control Software 1](#_Toc429645512)

[3.1 Definition 1](#_Toc429645513)

[3.2 Description 2](#_Toc429645514)

[4.0 Selection 3](#_Toc429645515)

[4.1 Repository Selected for Coursework 3](#_Toc429645516)

# Introduction

The purpose of this report is to describe repositories and their uses in Software Development, along with version control software. It also highlights the repository selected by me, for the duration of the OOP coursework, for the Fall 2015 term.

# Repositories

## 2.1 Definition

In general, the term repository defines a storage location. Software repositories provide storage for software packages which can be retrieved and installed onto a computer.

## 2.2 Purpose of Repositories

Repositories are used to store and distribute digital content. The main idea is to establish a location where content pertaining to a subject can be collected and then distributed to users related to that field. For software, repositories are ideal for collecting packages and files relating to single software can be stored. The use of repositories is ideal for open source software where hundreds of thousands of users can make changes to the core functionality, or add packages for added features. In these cases, repositories are the best method for keeping all the content in one place, where all the users can subscribe and keep track of all changes and new packages available. Repositories are available for many of the popular programming languages in use today. Some of them are listed below:

* **C++:** Boost
* **Java**: Maven and SpringSource
* **JavaScript**: Scripteka and JSAN
* **PHP**: Pear and Pecl
* **Perl**: CPAN
* **Python**: PyPi
* **R:** CRAN
* **Ruby**: RubyGems
* **TeX, LaTex**: CTAN

# 3.0 Version Control Software

## 3.1 Definition

Version Control Software is software which allows is the management of changes to documents, computer programs, large web sites, and other collections of information. Changes are usually identified by a number or letter code, termed the "revision number". Each revision is associated with a timestamp and the person making the change. Revisions can be compared, restored, and with some types of files, merged.

## 3.2 Description

Most software development is done in teams. As different people develop and add functionality simultaneously, different versions of the same software can exist in different locations. It is important to have a central collection where all changes and updates can be tracked and recorded. Also, during development, the latest changes may result in bugs, which would require the team to revert to a previous version of the code. Therefore, there is a need for a framework that can keep a record of all the changes made to the code, the person who made it, as well as access to the previous versions.

Since multiple team members may be working on the same document, often simultaneously, repositories allow users to branch the same document so that multiple copies are available. Users can then work on code in their own branches and then merge all of the changes into the main branch. This also highlights the ability in repositories to track all changes made on a file-by-file basis.

Some common terms and their definitions are given below:

* **Branch**: A set of files under version control may be branched or forked at a point in time so that, from that time forward, two copies of those files may develop at different speeds or in different ways independently of each other.
* **Change**: A change (or diff, or delta) represents a specific modification to a document under version control. The granularity of the modification considered a change varies between version control systems.
* **Checkout**: To check out (or co) is to create a local working copy from the repository. A user may specify a specific revision or obtain the latest. The term 'checkout' can also be used as a noun to describe the working copy.
* **Clone**: Cloning means creating a repository containing the revisions from another repository. This is equivalent to pushing or pulling into an empty (newly initialized) repository. As a noun, two repositories can be said to be clones if they are kept synchronized, and contain the same revisions.
* **Commit**: To commit (check in, ci or, more rarely, install, submit or record) is to write or merge the changes made in the working copy back to the repository. The terms 'commit' and 'checkin' can also be used as nouns to describe the new revision that is created as a result of committing.
* **Conflict**: A conflict occurs when different parties make changes to the same document, and the system is unable to reconcile the changes. A user must *resolve* the conflict by combining the changes, or by selecting one change in favour of the other.
* **Merge**: A merge or integration is an operation in which two sets of changes are applied to a file or set of files.
* **Pull, push**: Copy revisions from one repository into another. Pull is initiated by the receiving repository, while push is initiated by the source. Fetch is sometimes used as a synonym for pull, or to mean a pull followed by an update.

# 4.0 Selection

## 4.1 Repository Selected for Coursework

The requirement for the OOP course for the Fall 2015 term was to select and use a repository to upload all classwork and tasks to that repository. For that purpose, I have selected Github as the repository.

GitHub is a Web-based Git repository hosting service, which offers all of the revision control and source code management functionality of Git as well as adding its own features. GitHub provides a Web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project. GitHub is the largest repository service in the world and is used for hosting by a number of well know open source software projects. For these reasons, and its ease of use and interface, I selected Github as the repository for this course.

As stated earlier, GitHub provides an implementation of the Git architecture for version control. On the local machine, Git provides a command line interface, which allows a user to create local repositories from folders. These repositories can then be linked to the online ones on Github. Git also allows integration with Eclipse, and the changes to code can be directly uploaded from the developing environment.