# Gitfs Charter

V.Temian

Version 0.2.5

# PROJECT CHARTER

Vlad Temian

# GitFS

VERSION: 0.2.5 REVISION DATE: 13.01.2015

*Approval of the Project Charter indicates an understanding of the purpose and content described in this deliverable. By signing this deliverable, each individual agrees work should be initiated on this project and necessary resources should be committed as described herein.*

| Approver Name | Title | Signature | Date |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

# Contents

# Section 1. Project Overview

## 1.1 Problem Statement

gitfs was designed to bring the full powers of git to everyone, no matter how little they know about versioning. A user can mount any repository and all the his changes will be automatically converted into commits. gitfs will also expose the history of the branch you're currently working on by simulating snapshots of every commit.

## 1.2 Project Description

gitfs is a FUSE file system that fully integrates with git. You can mount a remote repository’s branch locally, and any subsequent changes made to the files will be automatically committed to the remote.

gitfs is useful in places where you want to keep track of all your files, but at the same time you don't have the possibility of organizing everything into commits yourself

## 1.3 Project Goals and Objectives

We need to have a fully fladge POSIX filesystems that will:

- Automatically commits changes: create, delete, update files and their metadata
- Browse through working index and commit history
- Merges with upstream by automatically accepting local changes
- Mounts the file system as a user or a group
- Caching commits reduces the memory footprint and speeds up navigation
- Reduces the number of commits by grouping pushes

## 1.4 Project Scope

| Project Includes |
|---|
| Automatically commits changes: create, delete, update files and their metadata |
| Browse through working index and commit history |
| Merges with upstream by automatically accepting local changes |

| Project Excludes |
|---|
| Data synchronization if the network is not reliable |

## 1.5 Critical Success Factors

## 1.6 Assumptions

- Python is a stable language
- Pygit2 is a maintainable technology

## 1.7 Constraints

- Deadline: 13.01.2015
- No budget
- Project needs to be open-source
- The technology should be Python
- Maximum of three dependencies
- Portable

# Section 2. Project Authority and Milestones

## 2.1 Funding Authority

*Free.*

## 2.2 Project Oversight Authority

*Vlad Temian is in charge of issues planing and time management.*

## 2.3 Major Project Milestones

*List the project's major milestones and deliverables and the planned completion dates for delivery. This list should reflect products and/or services delivered to the end user as well as the delivery of key project management or other project-related work products.*

| **Milestone/Deliverable** | **Planned Completion Date** |
|---|---|
| version 0.1 | Oct 13, 2014 |
| version 0.2 | Oct 27, 2014 |
| version 0.2.3 | Nov 13, 2014 |
| version 0.2.5 | Dec 12, 2014 |

# Section 3. Project Organization

## 3.1 Project Structure

The people that work on this project should be lead by a team leader that will be responsible for specifying the vision and make sure that it’s implemented. In order to keep track of the issues and implement the vision in a reasonable timeframe, a manager should project the issues and tasks in a manageable time frame. Developers should implement the tasks and tester will test the project in order to ensure the highest quality for the product.

## 3.2 Roles and Responsibilities

| **Role** | **Responsibility** |
|---|---|
| Lead | Describe the vision and milestone that should be achieve |
| Manager | Project in a manageable time frame the issues for each milestone |
| Developer | Resolve the issues |
| Tester | Ensure the highest quality for the product |

## 3.3 Project Facilities and Resources

*Describe the project's requirements for facilities and resources, such as office space, special facilities, computer equipment, office equipment, and support tools. Identify responsibilities by role for provisioning the specific items needed to support the project environment.*

| **Resource Requirement** | **Responsibility** |
|---|---|
| Drone.io | Continuous integration service |
| Github.com | Code hosting and issue tracker |
| Trello | Issue tracker |
| 3 VPS | Replicate a fully working environment |

# Section 4. Points of Contact

| Primary Contact | Name/Title/Organization | Phone | Email |
|---|---|---|---|
| Vlad Temian | Vlad Temian | +40740722577 | vladtemian@gmail.com |

# Section 5. Glossary

# Section 6. Revision History

*Identify document changes.*

| Version | Date | Name | Description |
|---|---|---|---|
| 0.1 | 15.10.2014 | Initial revision | Initial revision |
| 0.2 | 17.12.2014 | Pre-release | Adjust project structure and feature |
| 0.2.5 | 13.01.2015 | Final version | Complete all the reaming parts |

# Section 7. Appendices