SOMAIYA VIDYAVIHAR UNIVERSITY

Somaiya TRUST

K J Somaiya School of Engineering

Department of Computer Engineering
Course: DevOps

IA – 1: Case Study

Tool: Fossil

Name: Harshitha Shetty

Batch: C-3

Roll no.: 16010122339

Date: 25/09/25

INTRODUCTION

Fossil is a distributed version control system (DVCS) designed as a self-contained, all-in-one tool. Created by D. Richard Hipp, the developer of SQLite, Fossil combines version control, bug tracking, wiki documentation, and a web interface into a single executable.

While Git is the dominant version control tool in DevOps, Fossil provides a lightweight alternative that simplifies setup and integrates collaboration tools directly. This report demonstrates Fossil's capabilities as an alternative version control tool in the DevOps lifecycle.

Problem Statement

Version control is a fundamental stage in the DevOps pipeline. Git has become the industry standard, but it requires additional services like GitHub or GitLab for full project management. Teams seeking an integrated, lightweight solution may face challenges with Git's steep learning curve and reliance on multiple platforms.

Fossil addresses this gap by offering a holistic platform that combines version control with issue tracking, documentation, and a web UI—making it easier for teams to collaborate without external tools.

Objectives

- Demonstrate Fossil as an alternative version control tool in DevOps.
- Implement Fossil in a real-world workflow with repositories and commits.

- Compare Fossil with mainstream Git-based workflows.

Implementation and Demonstration

Installation (Windows):

- Download Fossil from https://fossil-scm.org.
- Place fossil.exe in C:\Fossil and add it to the system PATH.
- Verify installation: fossil version

Repository Creation:

```
fossil init myproject.fossil
mkdir myproject
cd myproject
fossil open ../myproject.fossil
```

Adding and Committing Files:

- Create a demo file index.html.
- Add and commit:

```
fossil add index.html
fossil commit -m "Initial commit"
```

Launching Web Interface:

```
fossil ui
```

This opens a local server at http://127.0.0.1:8080, where project timeline, issues, and wiki can be managed.

Comparison with Mainstream Tools

Aspect	Fossil	Git (Mainstream)
Ease of Setup	Single binary, no dependencies	Requires Git + GitHub/GitLab setup
Project Management	Built-in wiki, issue tracker, forum	External platforms needed
Collaboration	Autosync mode for distributed teams	Manual push/pull workflows
Learning Curve	Easier for beginners (web interface)	Steeper, command-heavy
Industry Adoption	Low (niche)	Very high, industry standard

Advantages & Limitations

Advantages:

- Lightweight and easy installation.
- All-in-one solution (VC + Wiki + Issues + Web UI).
- Autosync for seamless collaboration.
- Transparent project history and timeline.

Limitations:

- Low adoption and smaller community support.
- Limited CI/CD integrations.
- Not suited for large enterprise-scale projects.

Conclusion

Fossil demonstrates that alternative DevOps tools can provide strong value beyond mainstream options like Git. Its integrated platform simplifies project management by combining version control, bug tracking, and documentation into a single tool.

Through this case study, Fossil was successfully installed, configured, and tested with a demo project. The comparison with Git highlights its ease of use and integrated features, though industry adoption remains limited.

Fossil is best suited for small to medium teams who need a simple, lightweight solution with built-in collaboration tools. This exploration reinforces the importance of evaluating alternative tools in the DevOps ecosystem.