Can Git Repository Visualization Support Educators in Assessing Group Projects?

A presentation on the paper published in VISSOFT 2022

Presented by Felicia Sun

September 19, 2023

Introduction

- "Can Git Repository Visualization Support Educators in Assessing Group Projects?"
  - Mircea Lungu
  - Rolf-Helge Pfeiffer
  - Marco D'Ambros
  - Michele Lanza
  - Jesper Findahl

- 2022 Working Conference on Software Visualization (VISSOFT) in Limassol, Cyprus
- How software visualization tools can also prove to be invaluable assets in the realm of education.
- How these tools can aid educators in evaluating the quality of software systems developed by students.

Background

- The term “Software visualization tool”
  - Diagrams
  - Charts
  - Graphs
- Limited research about those tools in education
  - Exception: algorithm animation.
    - Sorting algorithms
    - BFS/DFS
- How git repository visualization tools can help educators in evaluating group projects
  - Commit history
  - Branch structures
  - Code contributions
  - Collaboration patterns
  - Code dependencies
- Context

Key differences

Educators evaluating large projects in a short time have a different context than other stakeholders supported by visualization tools.

Reverse engineers

Developers that are “onboarded”

Solo-developers

Technology or domain experts

Educator Needs

Unique requirements and constraints faced by educators:

- Need to assess large projects in a pretty short amount of time.
- Educator's time is limited
  - Cannot read complete solution
  - Source code of the project is only one of multiple deliverables
- Importance of evaluating individual contributions
  - Contributions can be unbalanced
- Teaching multiple courses with diverse technologies.
  - Supervise multiple thesis projects simultaneously
  - Need a technology-independent tool
- Privacy concerns for institutional repositories.
  - Often need to assess private or institutional repositories

Assumptions

And so it’s under those assumptions that this piece of research was carried out.

Git-Truck

- Researchers utilized git-truck, a visualization tool.
- Git-truck runs on personal computers from local git repository clones.
- https://github.com/git-truck/git-truck
- Git-truck's name is inspired by the "truck factor," which measures project continuity.

Usage Examples

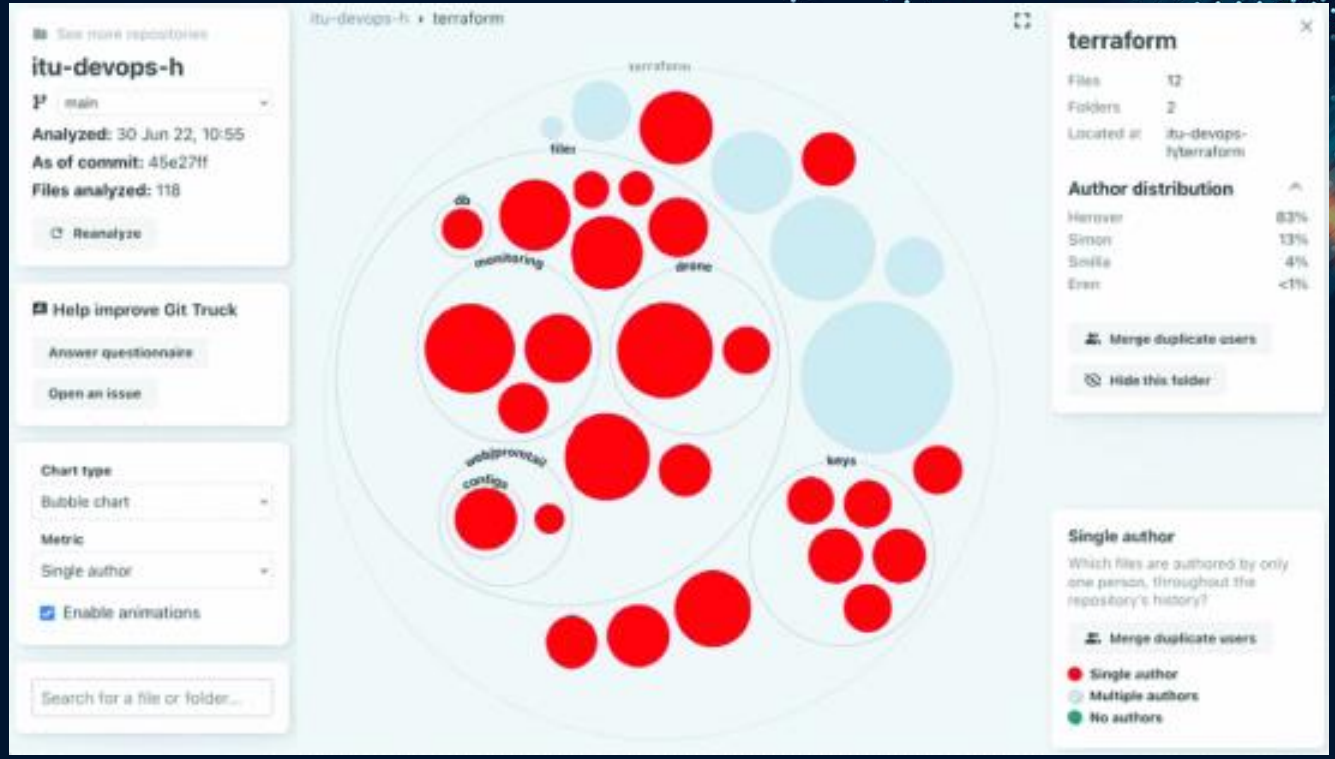
We’re going to look at some case studies showing the practical applications of a software repository visualization tool.

Examples were selected by the researchers in collaboration with educators from:

- IT University of Copenhagen (ITU) in Denmark
- Università della Svizzera italiana (USI) in Switzerland

Usage Example 1: Finding components with a single author

- Git-truck has a feature that highlights files that only had a single author in red
  - Lack of collaboration amongst students.
- Part of the source code of a group project
- Image zoomed into the terraform folder.
  - Terraform is an infrastructure-as-code tool.
- Almost all the work on this component has a single author
- Oral exam

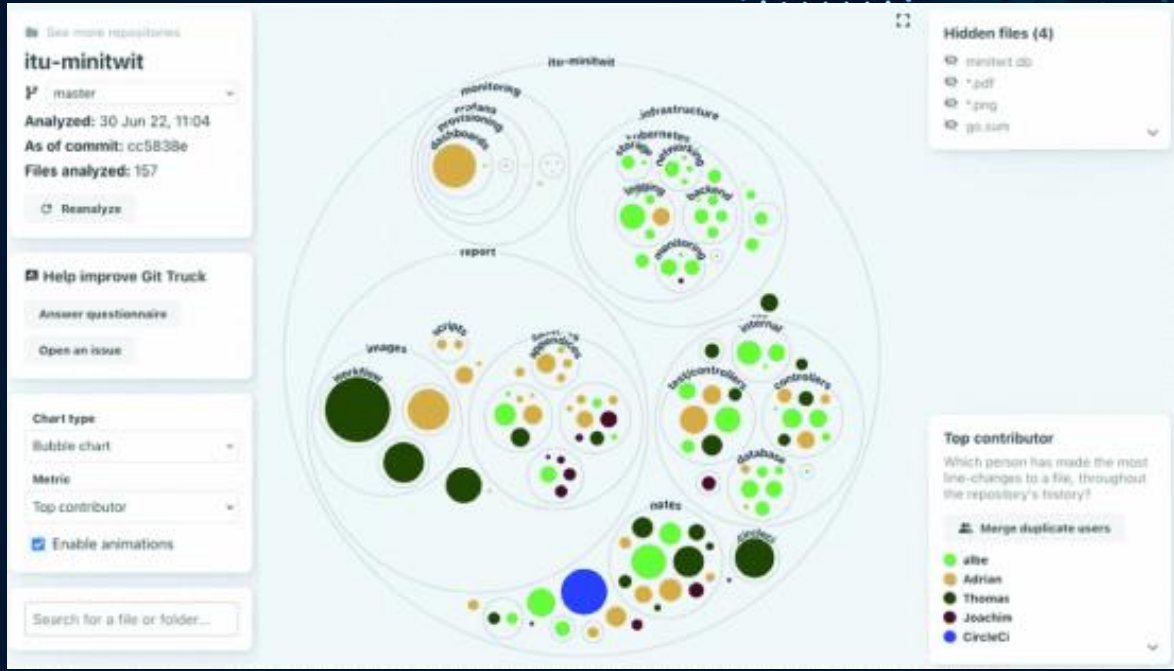


Usage Example 2: Investigating Responsibility Distribution in a Project

Feature:
- Color-codes top contributor for each file
- Identifies the author with the most lines added or removed in a file's history

Purpose:
- Assess distribution of work among project members

Example:
- Top Contributors view in another group's repository
- Highlights collaboration patterns within the project



Usage Example 3: Investigating Responsibility Distribution in a Project

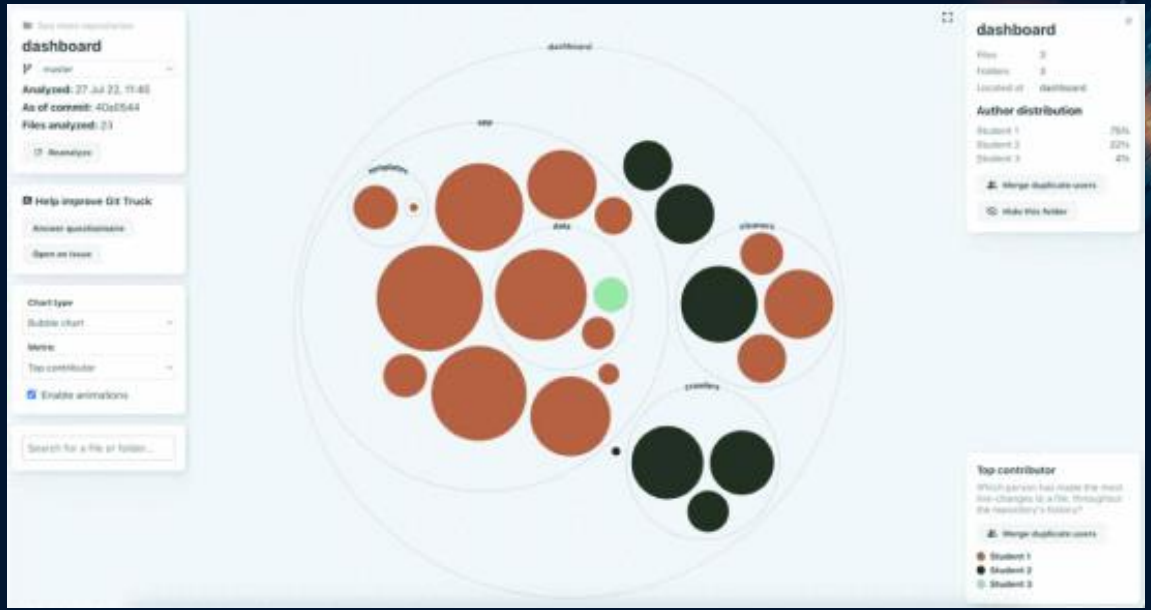
Scenario:
- Single author dominates most code files

Project goal:
- Experience composing a larger application from separately taught components
- Educator initiated discussion about individual contributions

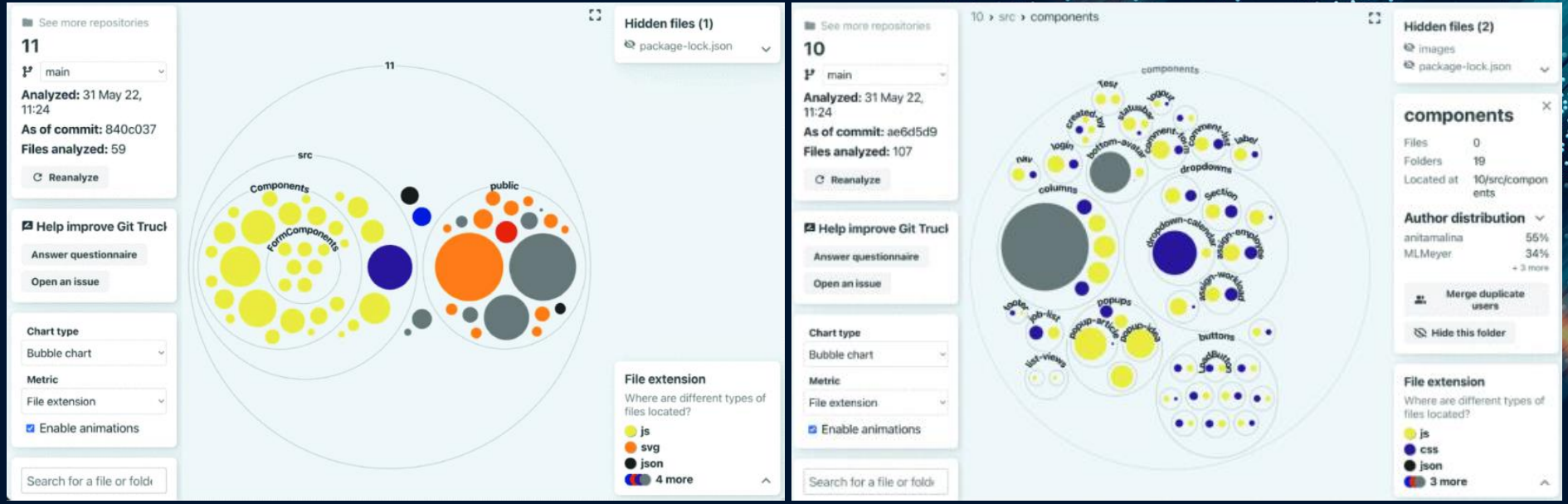
Discovery:
- Top contributor was a new Git user who accidentally deleted and restored the repository

Reminder:
- Caution against blind action based on visualizations; emphasize discussion

git-truck feature:
- Select last commit for analysis
- Useful for identifying collaboration issues arising after a specific commit



Usage Example 4: C. Gaining High-Level Architectural Insights



- Feature: Color-coding based on file extensions
    - Visual scaling based on file size and folder hierarchy
    - Useful for understanding systems with multiple languages
- Example: Two projects in an intro web-dev course
    - Yellow files represent JavaScript, purple files represent CSS
- Visual representation aids in witnessing architectural extremes
- Emphasizes the importance of discussing file organization in future course iterations

Usage Example 5:
D. Uncovering Critical Components of a System

Feature:

- Color-coding based on change frequency
- Darker colors = files with the most changes

Example:

- Author couldn't spot the "god class" without git-truck
- The "god class" handled user interface interactions, database querying, and scheduling
- Useful for identifying hidden bad design deep within the directory tree

Discussion: Limitations Of Metrics-Based Repository Visualization

- Caution against sole reliance on visualizations
  - confirm hypotheses with students or other sources
- Limitations include misleading metrics due to:
  1. Automatically generated code
  2. Committing files that shouldn't be tracked
- Unforeseen usage patterns
  - e.g. student removing and adding all repository contents
- Key takeaway

Discussion: Educator-specific Tool Support

After carrying out the experiment, the researchers compiled a list of 6 key functionalities that they believe are instrumental in increasing the adoption of such tools in education.

1. Interactivity
2. Author Unification
3. Configurable Thresholds
4. Automatic File Filtering
5. Integration of Commit Messages
6. Support for Multi-Repositories

Key functionalities to make the tools more user-friendly and tailored to the needs of educators.

Discussion: Generalizability

- Git-truck prioritizes usability in its design

- Visualizations implemented are not unique or challenging to replicate

- Authors suggest that the presented usage examples can benefit other educators using similar tools

Related Work

Wattenberger and Tornhill: Circle packing for repository metrics, developer and business focus

- No interactive features in their services

Raclet and Silvestre: Git4School, analytics dashboard for educators, commit-by-commit monitoring

Kim et al.: Githru, interactive git repository visualization, git commit graph focus, distinct from paper's objectives

Related Work

Cosentino et al.'s Gitana:
Computes truck factors, identifies crucial authors for software projects.

Gource (https://gource.io/):
Animates authors and their contributions over time.

Git Timeline Generator (https://www.preceden.com/git):
Visualizes contribution frequencies over time.

git-of-theseus:
Creates static visualizations of repository growth over time.

GitHub's built-in repository visualizations:
Presents activity statistics, such as commit frequencies and number of contributors, among others.

Conclusions and Future Work

- Educators as distinct user category for software visualization tools
- Multiple case studies across courses and universities
- Git repository visualization aids educators in group project assessment
- Need to expand research to include broader group of educators
- Goals: Solidify conclusions, offer guidance to educators and tool developers
- Explore extended usage of tools throughout the semester
- Understanding dynamic role of tools in education

THANKS!

References

M. Lungu, R. -H. Pfeiffer, M. D'Ambros, M. Lanza and J. Findahl, "Can Git Repository Visualization Support Educators in Assessing Group Projects?," 2022 Working Conference on Software Visualization (VISSOFT), Limassol, Cyprus, 2022, pp. 187-191, doi: 10.1109/VISSOFT55257.2022.00030.

A. Tornhill, "Your code as a crime scene: use forensic techniques to arrest defects bottlenecks and bad design in your programs", Your Code as a Crime Scene, pp. 1-218, 2015.

A. Wattenberger, "Visualizing a codebase", 2021, [online] Available: https://githubnext.com/projects/repo-visualization.

J.-B. Raclet and F. Silvestre, "Git4school: A dashboard for supporting teacher interventions in software engineering courses", European Conference on Technology Enhanced Learning, pp. 392-397, 2020.

Y. Kim, J. Kim, H. Jeon, Y.-H. Kim, H. Song, B. Kim, et al., "Githru: visual analytics for understanding software development history through git metadata analysis", IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 2, pp. 656-666, 2020.

V. Cosentino, J. L. C. Izquierdo and J. Cabot, "Assessing the bus factor of git repositories", 2015 IEEE 22nd International Conference on Software Analysis Evolution and Reengineering (SANER)., pp. 499-503, 2015.

A. Ciani, R. Minelli, A. Mocci and M. Lanza, "Urbanit: Visualizing repositories everywhere", 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)., pp. 324-326, 2015.

B. Scott-Hill, C. Anslow, J. Ferreira, M. Kropp, M. Mateescu and A. Meier, "Visualizing progress tracking for software teams on large collaborative touch displays", 2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)., pp. 1-5, 2020.