GitInsights

Rupesh Shantamurty, Divya C Nambiar, Avinash Upadhya, Vikrant Mah Dhimate

SDCG R&D

[rupeshs@hpe.com](mailto:rupeshs@hpe.com) , [dnambiar@hpe.com](mailto:dnambiar@hpe.com) , [avinash.upadhya@hpe.com](mailto:avinash.upadhya@hpe.com) , [vikrant.mah.dhimate@hpe.com](mailto:vikrant.mah.dhimate@hpe.com)

Abstract

In the entire development process for a software, code changes and it changes a lot. A lot of code gets written and modified on daily basis. Once the product matures engineers realise that some part of the software should have been architected and designed differently.With the ever changing structure of development teams it is difficult to retain the context of most of the changes done along the way. So the challenge development teams face is in identifying the components of the software that should be undertaken for redesign or change. To address this challenge we are proposing a framework in this paper which analyzes the trends in development lifecycle and presents the components that should be taken up for change. We call this framework GitInsights since it is based on Git revision control system. This framework will also assist software architects and product managers to make better decisions towards implementation of new features as well. Better decisions save time and expidite the product development lifecycle.

Problem statement

We at HPE write good software and that is the reason our software lasts for many years and sometimes lives for decades together. Such matured software undergoes multiple rounds of rearchitecting and redesign. It becomes important to decide on which components need to be chosen for the redesign and rearchitecting. The challenge is in identifying the components that are the trouble makes and that need change. Another challenge is to have enough data/justification for choosing the component.

Our solution

In this paper we present a framewok which consumes the revision control historical data and performs trend analytics on the captured data and presents it in a manner that is easily consumed by the end user. Below figure 1 show the basic design of our fremwork. Visualisation is done using D3 and the backend uses a combination of scripts and gitpython library.

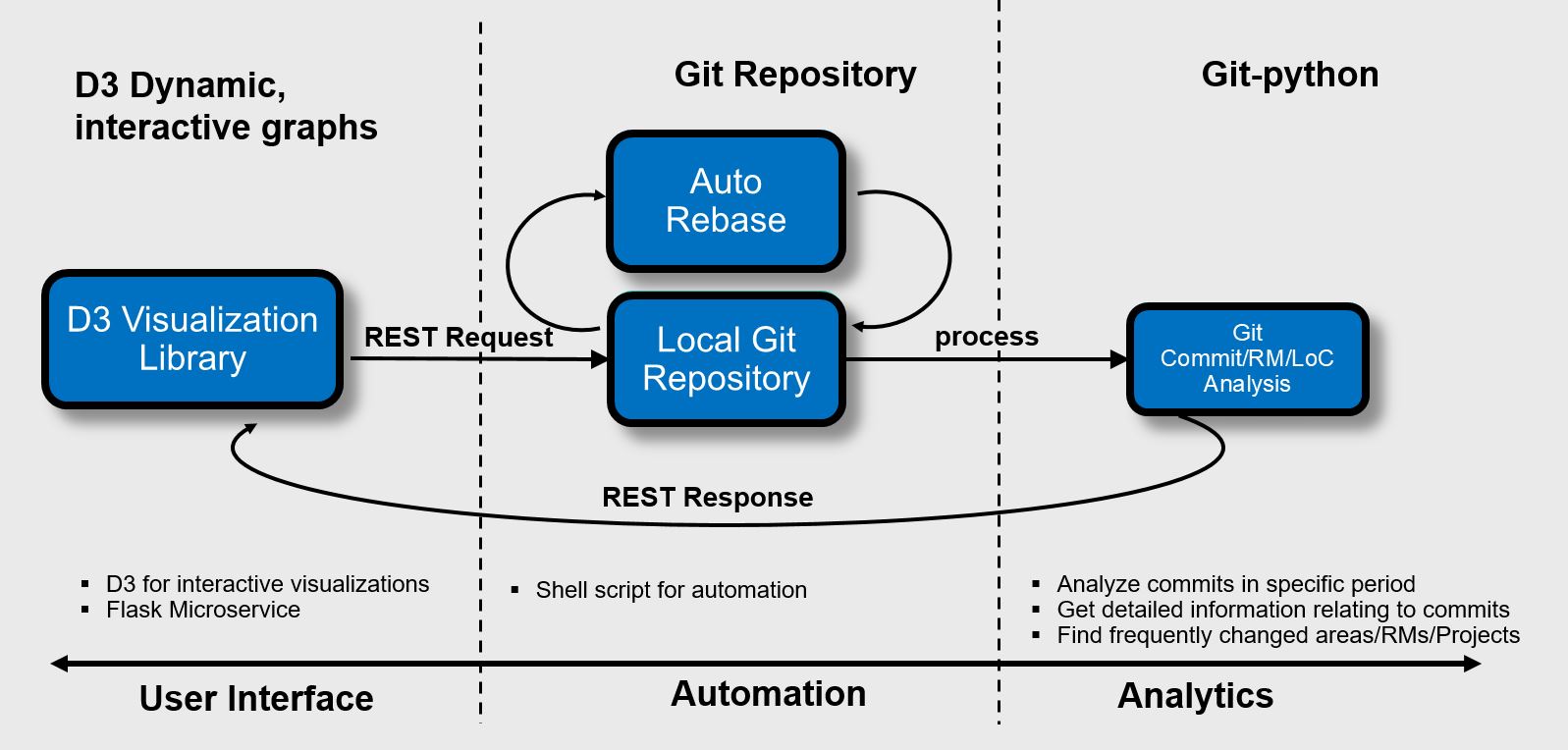


Figure 1. Implimentation of GitInsights

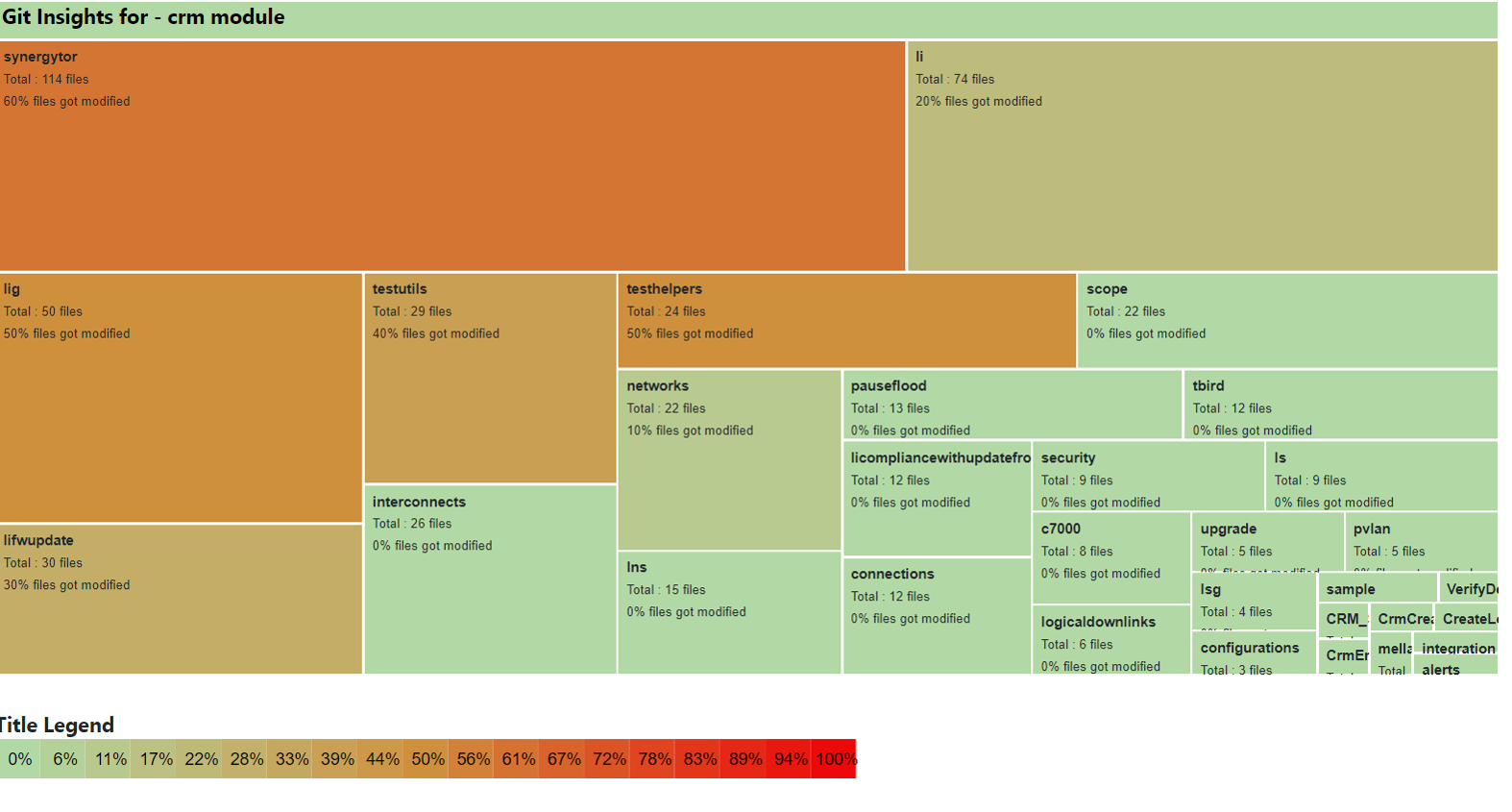
A interactive dashboard can be generated for identifying various trends like :

* Most modified modules represented as a heat map of the modules modified.
* Most frequently changed modules
* Changes to a module requires changes to many other modules 🡪 representing loose / tight coupling between modules
* Any custom requirements, yet to be identified

The code is generally organized in various folders/packages. The packages generally represent the implementation of a specific functionality. These represent the software components. The dashboard will be interactive in the sense that we are able to zoom in from a root view and drill down to sub levels of code hierarchy and are able to identify the final leaf components that got modified. The final leaf level modification is represented in term of number of lines changed through various commits in the git repository.

Evidence the solution works

We were able to quickly plug together the various modules as explained in the solution above and present a heat map of the changed module within two given dates. The figure 2 below is a screenshot of the implemented proof of concept :

Figure 2. Implimentaion of GitInsights showing heatmap of code changed between two dates

The above dashboard is clickable and that makes it interactive. One can double click and drill down to the files level where the dashboard represents the lines of code changed and the corresponding git commit information.

Competitive approaches

We found two instances where people have attempted to solve the problem presented here. One is GitStats – This tool visually presents the activities within a git repository. Our solution is different from GitStats in terms of not only presenting the data in a better interactive manner but also having capability of cutom trend analytics. Other prior art is a patent titled Software development automated analytics by Microsoft corporation. Our solution is different from this in terms of a simpler solution also the patent does not mention about the capability to customize the framework based on the business need. Our solution is customizable based on the requirements.

Current status

We were able to demonstrate a proof of concept implementation and our idea got awarded as one of the best ideas during the recently concluded OneView Hackathon Aug, 2019.

Next steps

We paln to provide this solution as a service and incorporate capability to generate more dashboard to show

1. Frequently modified modules.
2. Display modules that got changed together so as to identify the coupling between modules
3. Solicit inputs from the larger development and management community and incorporate more dashboards.

References

1. D3 Visualizations <https://d3js.org/> For frontend user interface
2. GitPython - <https://github.com/gitpython-developers/GitPython> For backend implimentation
3. GitStats - <http://gitstats.sourceforge.net/>
4. Software development automated analytics - <https://patents.google.com/patent/US8745572B2/>