8-14-2015

Software Metrics and Dashboard

Shilpika Shilpika
sshilpika@luc.edu

George K. Thiruvathukal
Loyola University Chicago, gkt@cs.luc.edu

Saulo Aguiar
Loyola University Chicago, saguiar@luc.edu

Konstantin Läufer
Loyola University Chicago, klaeufer@gmail.com

Nicholas J. Hayward
Loyola University Chicago, nhayward@luc.edu

Recommended Citation
Shilpika, Shilpika; Thiruvathukal, George K.; Aguiar, Saulo; Läufer, Konstantin; and Hayward, Nicholas J.. Software Metrics and Dashboard. , , , 2015. Retrieved from Loyola eCommons, Computer Science: Faculty Publications and Other Works,

This Presentation is brought to you for free and open access by the Faculty Publications at Loyola eCommons. It has been accepted for inclusion in Computer Science: Faculty Publications and Other Works by an authorized administrator of Loyola eCommons. For more information, please contact ecommons@luc.edu.

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 License.
Software Metrics Dashboard
Shilpika, Saulo Aguiar, Dr. George K. Thiruvathukal, Dr. Konstantin Läufer, Dr. Nicholas Hayward
sshilpika@luc.edu, saaguiar@luc.edu, gkt@cs.luc.edu, laufer@cs.luc.edu, ancientlives@gmail.com

About Software Metrics
- Computed from one or more measured values
- A critical tool that provides continuous insight to products and processes
- Helps build reliable software in mission-critical environments.
- The two types of metrics relevant to our work are:
  - Process metrics: Information that can provide insight into the underlying software development process.
  - Quality metrics: Information that can provide insight into the underlying software development process.

Approach
1. Evaluate whether CSE teams find the metrics dashboard useful.
2. Evaluate the effect of the metric dashboard on software quality and software process.
3. Implement dashboard based on metrics derived from information collected by the tools (GitLab and Bitbucket) used by projects.
4. We add new metrics as they become necessary.
5. We use Apache Spark, a cluster computing platform which serves as a general purpose engine for large scale data processing.

Preliminary Results
The metrics dashboard is developed using modern web development methodologies like Spray in Scala which provides client-server side REST/HTTP support on top of Akka.

Conclusions
Building appropriate sets of metrics, presented in a useful way, can prove beneficial to CSE software teams, large, small, and solo.

Facilitating the production of quality software would be a key component for developing and maintaining CSE software, especially as other mission-critical projects grow to depend on it.

In this project, we take steps to introduce a pragmatic set of metrics into CSE software projects by conducting surveys, building a metrics dashboard, and performing analysis and post-surveys on selected projects.

For interactive exploration of Metrics information and reduction in computation overhead the datasets are partitioned into clusters in a distributed environment which introduces concurrency and independent failures/recovery of partitioned tasks.

Future Work
We will ensure the metrics dashboard is properly instrumented to allow actual usage of the tools to be determined as projects collectively take advantage of them.

We plan to identify a set of metrics that are helpful to our own project (and for dissemination beyond scientific software teams in the future).

We will work toward a plugin framework, so teams can extend the dashboard with additional metrics we have not implemented yet.

Year 2 will end with a formal release of the metrics dashboard and online user manual.

Work Plan
The task requires us to perform the following activities:
1. Assess how metrics are used and which general classes/types of metrics will be useful in CSE (Computational Science and Engineering) projects.
2. Develop a metrics dashboard that will work for teams using tools like Github, Bitbucket etc.
3. Assess the effectiveness of the dashboard in terms of project success and developer attitude towards metrics and process.

Code Base
- https://github.com/LoyolaChicagoMetrics/loyolachicagometrics
- https://github.com/sshilpika/metrics-dashboard-commit-density
- https://github.com/sshilpika/metrics-dashboard-storage-service

Bibliography
- http://spray.io/introduction/spray-for-web-development/
- http://spray.io/introduction/spray-for-servers/
- Scala and Akka
- Akka Scala Documentation
- Spray toolkit for Scala and Akka

Acknowledgements: This material is based upon work supported by the National Science Foundation under Grant No. 1465347.