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

Follow this and additional works at: https://ecommons.luc.edu/cs_facpubs

Part of the Computational Engineering Commons, Computer Engineering Commons, and the Computer Sciences Commons

Recommended Citation
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 work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 License.
Software Metrics Dashboard
Shilpika, Saulo Aguilar, Dr. George K. Thiruvathukal, Dr. Konstantin Läufer, Dr. Nicholas Hayward
sshilpika@luc.edu, saaguilar@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:
  - Computational metrics: intrinsic code properties like code complexity
  - Governance 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 (Griffith 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.

The metrics dashboard will continually compute the metric values and expose web services that allows developers to obtain metrics reports various textual and graphical formats.

We use Apache Spark, a cluster computing platform which serves as a general purpose engine for large scale data processing.

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 useful for 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 they have not implemented yet.

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

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 useful for 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 they have not implemented yet.

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

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

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