# Lightweight, automated process and product metrics collection and analysis with Hackystat

Philip M. Johnson

Collaborative Software Development Laboratory
Department of Information and Computer Sciences
University of Hawai'i
Honolulu, HI 96822, USA
johnson@hawaii.edu

Hackystat is an open source framework for automated collection and analysis of software engineering process and product data that has been under development since 2001. Increasing usage by a variety of academic and professional organizations indicates its utility to organizations interested in lightweight, yet sophisticated metrics collection and analysis.

Hackystat differs from other metrics collection and analysis frameworks in one or more of the following ways.

First, Hackystat uses sensors to unobtrusively collect data from development environment tools; there is no chronic overhead on developers to collect product and process data. Over two dozen sensors are publically available, including sensors for IDEs (Emacs, Eclipse, JBuilder, Vim, VisualStudio), configuration management (CVS, Subversion), bug tracking (Jira), testing and coverage (JUnit, CppUnit, Emma, JBlanket), system builds and packaging (Ant), static analysis (Checkstyle, PMD, FindBugs, LOCC, SCLC), and so forth.

Second, Hackystat is tool, environment, process, and application agnostic. The architecture does not suppose a specific operating system platform, a specific integrated development environment, a specific software process, or specific application area. A Hackystat installation is configured from a set of modules that determine what tools are supported, what data is collected, and what analyses are run on this data.

Third, Hackystat is intended to provide in-process project management support. Many traditional software metrics approaches are based upon the "project repository" method, in which data from prior completed projects are used to make predictions about or support control of a current project. In contrast, Hackystat is designed to collect data from a current, ongoing project, and use that data as feedback into the current project.

Fourth, Hackystat provides infrastructure for empirical experimentation. For those wishing to compare alternative approaches to development, or for those wishing to do longitudinal studies over time, Hackystat can provide a low-cost approach to gathering certain forms of project data.

Fifth, Hackystat is open source and is made available for no charge.

Hackystat has been applied in a variety of agile and non-agile contexts. The Zorro Project involves the development of a sensor for Eclipse that attempts to automatically detect when users are employing test-driven design practices. An pilot validation study found that Zorro correctly identified TDD episodes 89% of the time [1]. Software Project Telemetry is an approach to metrics analysis and visualization that provides a new style of in-process project management [2]. Hackystat is being applied as a measurement technology for the DARPA High Productivity Computing Systems program [3]. Finally, Hackystat has been used extensively for software engineering education [4].

Developing proficiency with the Hackystat Framework involves a progression through three levels of expertise. The first level is the "user", who is able to install sensors and run the analyses provided by a server. The next level is the "administrator", who is able to install and maintain a Hackystat server, as well as assemble new configurations of Hackystat from the 70 publically available modules to customize the system's capabilities for their organizational needs. The final level is the "developer", who can implement new modules to support additional process or product data and analyses for entirely new software development domains.

The goal of this tutorial is to provide attendees with a clear understanding of the opportunities and challenges associated with Hackystat-based software engineering measurement collection and analysis. This understanding can help attendees decide whether or not Hackystat is appropriate for their organization, as well as improve their ability to evaluate software engineering metrics frameworks in general.

The tutorial will work toward this goal in a highly experiential way. Attendees are encouraged to bring a laptop (Unix, Mac, or Windows), upon which they will install a Hackystat server and sensors, collect their own sample metrics, and perform analyses on them. These direct experiences will be augmented by case studies on large scale usage of Hackystat.

## References

1. Kou, H., Johnson, P.M.: Automated recognition of low-level process: A pilot validation study of Zorro for test-driven development. In: Proceedings of the 2006 International Workshop on Software Process. (2006)
2. Johnson, P.M., Kou, H., Paulding, M.G., Zhang, Q., Kagawa, A., Yamashita, T.: Improving software development management through software project telemetry. IEEE Software (2005)
3. Johnson, P.M., Paulding, M.G.: Understanding HPCS development through automated process and product measurement with Hackystat. In: Second Workshop on Productivity and Performance in High-End Computing (P-PHEC). (2005)
4. Johnson, P.M., Kou, H., Agustin, J.M., Zhang, Q., Kagawa, A., Yamashita, T.: Practical automated process and product metric collection and analysis in a classroom setting: Lessons learned from Hackystat-UH. In: Proceedings of the 2004 International Symposium on Empirical Software Engineering, Los Angeles, California (2004)

## 1 Agenda

08:00-08:45: Overview of Hackystat

08:45-09:30: Lab 1: Installing Hackystat sensors, collecting initial data

09:30-10:00: Break

10:00-10:45: Basic Hackystat data analysis Project summaries and telemetry:

10:45-11:30: Lab 2: Basic analysis with Hackystat

11:30-12:00: Advanced analysis with Hackystat (TDD inference)

12:00-01:00: Lunch

01:00-01:30: Lab 3: Installing and configuring a Hackystat server

01:30-02:00: Hackystat and metrics data administration

02:00-02:30: Lab 4: Requirements for software engineering metrics in your organization

02:30-03:00: Break

03:00-03:30: Customizing Hackystat for specific organizations (JPL)

03:30-04:00: Roadmap for Hackystat 2007-2008:

04:00-whenever: Open questions and answers