MSR 2004
International Workshop on Mining Software Repositories
(25th May 2004, co-located with ICSE 2004)
"http://msr.uwaterloo.ca"
Call for Papers

Organizers

Ahmed E. Hassan
(aeehassa@plg.uwaterloo.ca)
Richard C. Holt
(holt@plg.uwaterloo.ca)
School of Computer Science
University of Waterloo
Ontario, Canada

Audris Mockus
(audris@research.avayalabs.com)
Software Technology Research Dept.
Avaya Labs Research
NJ, USA

Program Committee

Harald Gall (U. of Vienna, Austria)
Les Gasser (U. of Illinois, UC, USA)
Daniel German (U. of Victoria, Canada)
James Herbsleb (CMU, USA)
Katsuro Inoue (Osaka U., Japan)
Philip Johnson (U. of Hawaii, USA)
Dewayne Perry (U. of Texas, USA)
Andreas Zeller (Saarland U., Germany)

Important Dates

Intent to submit:
    27th February 2004
Deadline for submission:
    8th March 2004
Paper Notification:
    29th March 2004
Final papers due:
    12th April 2004
Workshop date:
    25th May 2004

Location

Co-located with ICSE 2004

Edinburgh International
Conference Centre, Scotland, UK

Website

http://msr.uwaterloo.ca

Software repositories contain a wealth of valuable information for empirical studies in software engineering: *source control systems* store changes to the source code as development progresses, *defect tracking systems* follow the resolution of software defects, and *archived communications* between project personnel record rational for decisions throughout the life of a project. Such data is available for most software projects and represents a detailed and rich record of the historical development of a software system. Participants in multiple sites, often in multiple continents, develop software projects without ever meeting in person, as is the case in many large commercial and Open Source projects. This trend makes the use of tools to record all aspects of software project more critical.

Until recently, data from these repositories was used primarily for historical record supporting activities such as retrieving old versions of the source code or examining the status of a defect. Several studies have emerged that use this data to study various aspects of software development such as software design/architecture, development process, software reuse, and developer motivation. These studies have highlighted the value of collecting and analyzing this data. Yet each of these studies has built its own version of methodologies and tools to address the formidable challenge of utilizing such data to perform their empirical research. Several international efforts have identified the development of approaches to extract, share, and study this data as a research priority.

The goal of this one-day workshop is to bring together researchers, and practitioners to consider methods to use data stored in software repositories to further understanding of software development practices. We expect the presentations and discussions in this workshop will facilitate the definition of challenges, ideas and approaches to transform software repositories from static record keeping repositories to active repositories used by researchers to gain empirically based understanding of software development, and by software practitioners to predict and plan various aspects of their project.

Position papers should be at most 5 pages and may address issues including but not limited to the following:
- New approaches to analyze the data stored in software repositories to:
  - Assist in program understanding and visualization
  - Predict and gauge the reliability and quality of software systems
  - Study the evolution of software systems
  - Discover patterns of change and refactorings
  - Understand the origins of code cloning and code design change
  - Model software processes for development, defect repair, etc.
  - Assist in project planning and resource allocation
- Case studies on extracting data from repositories for large long lived projects
- Proposals for exchange formats, meta-models, and infrastructure tools to ease the sharing of the extracted data and to enable reuse and repeatability of results throughout the community
- Suggestions for particular large software repositories to be shared among the community for research evaluation and benchmarking purposes
- Approaches to integrate data between repositories and with other software project data such static or dynamic analysis data
- Requirements and guidelines for users and developers of source control systems to ease the analysis of the stored historical data