

Mining Software Repositories

Daniel M German

Dept of Computer Science, University of Victoria

04 January 2010

1 MSR Introduction to the Course

Objectives of the course

- 1 To understand how Software Repositories can be used
- 2 To become aware of current research in the area
- 3 To create a project/report with the potential to be published
- 4 To learn to do research and write papers by reading papers

Web site:

- 1 Course website located at: <http://turingmachine.org/msr>

Outline of grading

- 1 Every person will present 2 papers, one assigned and one selected from a pool of papers
- 2 2 assignments in teams of 2 people. Deadline for selecting a team Monday 11 January
- 3 1 project/paper per person, several milestones, but major deadline end of term
- 4 **You are expected to attend all classes**

Breakdown

- 10% Classroom participation
- 20% paper presentations (2)
- 10% Weekly critique
- 20% 1 or 2 Assignments
- 40% Project

Mechanics of the course

- 1 Meeting twice a week: Monday and Wednesday
- 2 Critique's deadline is Monday at the beginning of class
- 3 Presentations: as scheduled
- 4 Assignments: 2 weeks after distributed
- 5 Term project/paper: end of term with ongoing deliverables

Reading assignment for next Monday

- 1 The Road Ahead for Mining Software Repositories by Ahmed E. Hassan (must critique)
- 2 The Future of Mining Software Archives by several

What is in critique?

- 1 One page.
- 2 Contains:
 - Brief summary
 - List of major contributions (I want quality, more than quantity)
 - At most 3 positive points about the paper
 - At most 3 negative points about the paper

Papers for first round:

- 1 Mining version histories to guide software changes, by Zimmermann et al.
- 2 What's Hot and What's Not: Windowing Developer Topic Analysis? by Hindle et al.
- 3 Hipikat: recommending pertinent software development artifacts, by Cubranic et al.
- 4 Analysing Software Repositories to Understand Software Evolution, D'Ambros et al.
- 5 Visualizing software changes, Eick et al.
- 6 Seeking the source: software source code as a social and technical, artifact de Souza et al.

Papers for first round (cont).

- 7. Automatic identification of bug-introducing changes by Kim et al.
- 8. Identifying reasons for software change using historic databases by Mockus et al.
- 9. Scalable statistical bug isolation by Liblit et al.
- 10. Measuring the Progress of Projects Using the Time Dependence of Code Changes, by Alamet al
- 11. Using Software Dependencies and Churn Metrics to Predict Field Failures: An Empirical Case Study Nagappan et al.

Scheduling

- I have to attend 2 workshops: we need to reschedule

Other questions?