Dear Scientist, Can We Help?

Luke Nguyen-Hoan
luke.nguyen-hoan@anu.edu.au

School of Computer Science, College of Engineering & Computer Science, Australian National University

What’s this for?
*Help identify areas where improvements in scientific software practice can be made
*Where and how can software engineering help?
*Part of my research which aims to help scientists do more science

The Survey
*Conducted online through the ANU’s Polling Online (APOLLO) system
*Distributed via email and mailing lists
*Run from August to September in 2009
*60 responses

Testing and Verification: there’s room for improvement
*Two of the three most widely performed activities involve comparison with other data
*Verification is the least performed activity
*More people verify against requirements than produce requirements documentation!

Tools are not widely used
*10 respondents said they used no tools
*IDEs and Version Control were the most commonly used, by around 50% of developers
*Still a lot of improvement possible!

Documentation is written... sometimes
*Comparative lack of documentation for requirements and design
*User manuals/guides quite common
*Not every developer comments their code

Documentation: It’s not worth the effort
*Time, effort, and cost to benefit ratio the most common hurdles for the production of documentation
*4 respondents believe documentation is unnecessary!

Reliability and Functionality are the most important
*Contradicting previous studies, portability is not considered important
*Traceability has low importance for developers, but previous studies have shown it is important for users

Most scientific software developers do not have a computing background
*Only 13 respondents have an educational background in software engineering or computing
*Note that some respondents listed more than one field of study