Improving CSE Software via Rigorous Research Expectations

Michael A. Heroux
Sandia National Laboratories
Albuquerque, NM
maherou@sandia.gov

8 Oct 2013

1 Introduction

Adoption and adaptation of modern software engineering (SE) processes and practices by computational science and engineering (CSE) software projects tends to be behind other SE areas. This issue is a concern for funding agencies, since new research increasingly relies upon and produces computational tools. Furthermore, computational results are increasingly used in decision support and design, where correct results are crucial.

At the same time, CSE funding organizations find it difficult to prescribe formal SE practices for funded projects. This is because there are no widely known or accepted software engineering approaches in the CSE community, and CSE project teams are not directly rewarded for investing in software engineering improvements. Furthermore, the historical success of formal software engineering processes applied to CSE projects is poor.

Experimental sciences rely heavily on independent verification of results as part of the scientific process. Furthermore, these scientists are expected to know the state-of-the-art tools and processes available to them, and justify that their tools, processes and practices are reliable. They rely heavily on independent confirmation of results as part of the scientific process.

In principle, computational science should have the same regard for independent confirmation, but a study of the literature shows that in many cases there is very little hope of reproducing results from the published information. Furthermore, results are often generated by software that is not managed, and it would be difficult in the future to identify the exact version of software that generated the results.

In this paper, we present an argument for using reliability, reproducibility and independent verification as drivers to improve SE processes and practices. Rather than prescribing specific software practices, tools and processes, which can often be interpreted as an imposition on otherwise autonomous groups, we propose to set expectations that are perhaps more natural for a research team to accept. Through these expectations, we obtain the desired software engineering improvements in software engineering and permit teams to be creative about satisfying the requirements. Furthermore, we observe that, although funding agencies may find it difficult to impose formal software engineering practices on funded project, it is very reasonable to expect research teams to obtain independent verification of their results and justify why we should trust their software tools. In fact, it is our opinion that if reliability, reproducibility and independent verification are expected, we will see rapid adoption and adaptation of software engineering practices in the research community, not because formal process were imposed but because researchers will need their results to be trusted and will want to minimize the overall effort that goes into achieving reproducibility and independent verification. Good software engineering is a part of the process.

2 Research Expectations

If a research team is expected to demonstrate that their software is reliable, and be able to readily reproduce computational results and support independent verification, several things become immediately useful:

1. **Source management tools:** In order to guarantee that results can be reproduced, the software must be preserved so that the exact version used to produce results is available at a later date.

2. **Use of other standard tools and platforms:** In order to reduce the complexity of an environment, standard software libraries and computing environments are attractive.

3. **Documentation:** Reliability and independent verification requires that someone else understand how to use your software.

4. **Source code standards:** Improves the ability of others to read your source code.

5. **Testing:** Investment in greater testing improves reliability, and usability by others.

6. **High-quality software engineering environment:** If a research team is serious about producing high-quality, reproducible and verifiable results, it will want to invest in a high-quality SE environment to improve team efficiency.

7. **Knowledge of best software engineering practices, processes and tools:** Knowledge of the state-of-the-art in SE practices becomes valuable to both demonstrate reliability and inform ongoing improvements in software engineering.

Requiring reliability, reproducibility and independent verification does have many positive influences on SE practices and processes. In fact, we believe that reliability, reproducibility and verification requirements are the most natural way help CSE teams make progress against SE goals.

3 Conclusions

CSE has often been described as the third fundamental paradigm for scientific advance, and it is truly recognized as a valuable pursuit, even being mentioned in two recent State of the Union addresses by US president Barack Obama. However, as a scientific process, the lack of reliability, reproducibility and independent verification is noticeable, as is the lack of software engineering rigor. We believe that these weaknesses are linked, and that demanding reliable, reproducible computational results will positively impact adoption and innovative adaptation of software engineering practices and processes in CSE.