Research Software Science: A New Approach to Understanding and Improving How We Develop and Use Software for Research

Introduction

Development and use of software are fundamental to many areas of scientific research. Many scientists write, modify and use software to gain insight and prove scientific results. At the same time, formal software engineering techniques and knowledge that are widely used in other software development domains are not as commonly used in research software projects. Many factors contribute to the apparent gap between practices in research software relative to other software communities providing opportunities for improvement.

In this paper, we propose that the scientific method—which is central to scientific efforts using research software—can be used to study and improve the development and use of research software. In other words, research software development and use are the subject of our scientific study.

Science Applied to Research Software

The primary objective of research software science as we are proposing it is to *apply the scientific method to understanding the development and use of research software.* This pursuit has strong social and technical components:

- **Technical component:** The purpose of research software is modeling and simulation of scientific theories, gathering, analyzing and understanding scientific data or similar pursuits, which typically require a deep understanding of the scientific domain and years of education in the domain. Someone without a conceptual understanding of the scientific area will not be able to understand the nature of the software.
- **Social component:** Development and use of scientific software is typically a team effort, increasingly involving more people and more diverse roles. Team interactions, workflows and tools play a large role in the effectiveness of a research software team. While many people have realized the importance of the technical component of research software development and use, fewer people have focused on the social elements and even fewer have applied a scientific approach to studying and improving research software team interactions.

Social Science Focus

Applying the scientific method to research software teams necessarily involves the social sciences. Observations, interviews, data mining and similar techniques provide the raw materials for analyzing and gaining understanding into important correlations and, hopefully, cause and effect, between behaviors, situations and outcomes.

Why Scientist and Not Engineer

Software engineering is a large field with many practitioners. However, as Fred Brooks has stated, “An engineer learns in order to build.” This approach does not provide adequate time for transforming the results of engineering best practices studies into scientific outcomes. Surely, there are good software engineering researchers who have published many studies, but then this is science, not engineering and we should call it science.

Why Now: Multi-disciplinary Direction of Science

Many important efforts in science require strong multi-disciplinary teams. We see that research software is increasingly multi-scale, multi-physics or involving modeling, simulation and data analysis. Adding the scientific pursuit of understanding research software development and use results in one more dimension in the pursuit and is especially appropriate given the growing diversity of scientific teams and the need to understand and optimize team interactions and output.

XYZ Software Science

One could argue that the scientific method could be applied to any software development and use community. This is true. In my own community, we often focus on high-performance computing (HPC) software as a special subclass of research software. Putting particular adjectives in front of software science helps to narrow the focus of study, perhaps leading to more specific outcomes.

Summary

Research software development and use is a rich and dynamic pursuit worthy of scientific study in its own right. Viewing this effort as a scientific problem opens the door to applying our skills as scientists to assist in making our software development and use even more effective. Forming and promoting the role of a research software scientist seems to be very promising as a way to improve science overall.