Software Engineering for Science

Better Software, Better Research”—that’s the motto of the UK’s Software Sustainability Institute (http://software.ac.uk) directed by my colleague Neil Chue Hong. While I didn’t invent the slogan, I really like it and think it communicates an important message. With the ever-increasing importance and diversity of scientific software also comes the increasing need for that software to be correct, maintainable, and sustainable. An important goal of software engineering is to provide tools, techniques, and practices to address these concerns. In my role as associate editor in chief of CiSE and with my background in software engineering, I see the opportunity to write this column as a chance to encourage the community to move in a positive direction.

In thinking about the important role for software engineering in the development of scientific software, I like to make an analogy with more traditional research instruments. Traditionally, scientists have relied on various types of scopes, meters, and other “tools of the trade.” A quick Internet search turns up numerous results related to “validation of XXX” (where XXX is some type of scientific tool). Contrast that with the way many scientists treat the validation of their software (one of today’s important “tools of the trade”)—those same scientists who wouldn’t think of using a physical instrument that wasn’t properly validated and calibrated will often use the results of a software package that has undergone a much-less-than-thorough validation process. While much of the resulting software might be correct, without proper validation of the software itself and its development process we can’t be confident of this as fact. A scientist could be staking his or her reputation on a faulty instrument. Such a situation could be tragic.

Addressing this problem is where the discipline of software engineering is relevant. Contrary to what many in the scientific community may think, the software engineering doesn’t necessarily have to refer to large-scale, rigid, documentation-heavy processes. Rather, it can refer to much smaller, lightweight practices that can provide both immediate and long-term benefits to scientific code without resulting in large expense or delay. My experience has been that many who are resistant to the use of software engineering might be unaware of these smaller-scale practices and the types of benefits they can provide. As a result, my current research interest is to work directly with scientific software teams to help them identify aspects of their software or their software development process that could use some improvement (and as anyone who has written software knows, we all have room for improvement!). The next step is to tailor an existing software engineering practice to work in the new context or develop a new practice, then deploy that practice and observe the results. By objectively and empirically documenting the successes (and failures) of the use of software engineering practices in science, we can begin to build up a body of knowledge that can benefit the whole community. If you’re interested in contributing to this research, please contact me at carver@cs.ua.edu. I’m always looking for collaborators to expand the potential impact of this work.

I also want to advertise two venues for publishing experiences with using software engineering in science. These venues should be of interest both to those who wish to share experiences and to those who are interested in learning more about the experiences of other scientists. First, I, along with other co-organizers, have been running the Software Engineering for Science workshop series for 10 years. In addition to providing a venue for presenting experiences and research results, these workshops offer...
an opportunity for attendees to spend time in smaller group discussions with other interested members of the software engineering and science communities. Find out more information about this workshop series along with links to previous and upcoming workshops at http://SE4Science.org/workshops. Second, there is a new Software Engineering track within Computing in Science & Engineering, edited by Damian Rouson and me. More details about the types of papers that fit this track can be found at www.computer.org/cms/Computer.org/ComputingNow/docs/2016-software-engineering-track.pdf.

Finally, I want to make a personal appeal to those CiSE readers who are developing software. In the traditional software engineering domain, individuals and teams use various types of metrics and data to track their process, identify needs, and measure improvements. To the best of our knowledge, we don’t have detailed information about the use of metrics in scientific software development. To gather this information, George K. Thiruvathukal (CiSE’s editor in chief) and I have created a survey about software metrics in science that takes only 10 or so minutes to complete: https://goo.gl/6qDmNZ. Feel free to share the link with anyone else who might be interested, too. We hope to use the results of this survey to inform the development of analytics tools for measuring software quality in scientific software.