Teaching SRE to Software Practitioners

John D. Musa
Independent Consultant
j.musa@ieee.org

Abstract

Teachers can learn many lessons by examining experiences in teaching SRE to software practitioners and analyzing their feedback. I have taught several thousand participants over 15 years in a wide variety of organizations and applications. Participants included software developers, testers, reliability and quality control staff, and managers. Most of what I’ve learned applies to any course and is not SRE-specific. It also applies to the practitioners of the future, university students.

1. Environmental constraints

The most important issue is the high-pressure demands of work in a competitive environment. These demands make courses longer than 2 days impractical; registrations fall off drastically for longer courses, so one must organize the material to fit within this limit. Semester length courses appear to work only when the participants have the motivation of working for a degree. Conflicts with work, urgent situations, meetings, etc. mean that a large percentage of class (often 20% to 30%) misses some class hours (often 10% to 25%). Organizations expect participants to bring back and implement new ideas at work; you need to increase the likelihood that this will happen. Finally, some audiences have participants of limited proficiency in English.

2. Resulting teaching requirements

Try to minimize class time missed by consulting in advance with the class about scheduling start, stop, lunch, and break times. However, since some missed time is unavoidable and since some participants may have difficulty with English, alternate means of receiving the course material are essential. Thus, there must be a book that is closely coordinated with the course and there must be detailed slides, with copies of both provided to all participants. Reinforce the material presented by applying it in workshops related to the participants’ projects.

The instructor should encourage participants to interrupt the class at any time to clarify issues of general interest. Suggest deferring individual issues (for example, from those who missed part of class) for one-on-one discussions at break periods.

If most participants have limited proficiency in English, slow the course’s pace and avoid humor and slang. Allow workshops in the participants’ native language, but have all decisions written in English on flip charts so the instructor can guide the workshop.

Practitioners prefer to learn an organized, tested process rather than a collection of techniques and tools. References to users, especially those who have written up their experiences, is important. The two-day constraint requires that you present a process that works for, say, 80% but not all projects. Separate the material for special situations.

Most practitioners have little interest in the theory that supports a practice, except to know its existence. Hence the course book should separate any theory from the core material but provide references. The course book should include FAQs collected from former course participants; they resolve the difficulties participants most commonly encounter.

To help implement the material taught in the workplace, provide information on technology transfer. Conduct a final workshop that addresses how the participants will apply what they have learned on the job, setting up action items, persons responsible, and target completion dates.

Some lessons learned are SRE-specific. Participants usually find the Engineering Just Right Reliability and Preparing for Test activities the most difficult to learn. Teaching how to use CASRE in a computer classroom was not worth the time, since there is a good user manual. In application, the importance of carefully defining product, customers, and users is usually grossly underestimated. The difficulty of determining occurrence rates for operations is usually overestimated.