Brief summary of the round table discussion held at FormaliSE 2013

The discussion mostly revolved around the challenges in teaching Software Engineering and the need to apply formal method techniques for the same. At the undergraduate level, every now and then, we see a course on formal methods but that has not been pervasive right through. Perhaps, we do not see Software Engineering as engineering. Engineering should primarily protect the public from harm. There's a need to be a little bit more careful about what we call software engineering. A number of points came up during the discussion:

- Introducing formality everywhere risks reducing the fun out of certain tasks like programming etc. It hampers the creative motivation. We have to be smart about how to get people interested in formal methods.
- It is important to do something real to get excited. Examples that we give in courses, most of the times, have nothing to do with the real world. The motivation has to be real.
- The fun side of engineering is building things that are of practical value and importance (for instance, building a pace-maker). Besides, students think that such experiences would add more value to the resume. As a community we can do several things that are challenging and interesting and, at the same time, is of critical value.
- Sometimes there's a need to package an entire course on formal methods into a few lectures by focusing on a particular tool, or a language, and focusing only those concepts which are important for that. Software companies aren't ready to send their best engineers for a two weeks course. Three-day courses are still acceptable. Educators should, perhaps, also think about this. Besides, a group of people who have to be taught this way may have different backgrounds and different levels of motivations. Educators need to work around these.
- The short courses should be such that it is enough to get started in that field. And the rest of the learning can happen gradually. One wouldn't be done after a three-day course, but it's an acceptable beginning. After that, a lot would come through observing what is happening and asking questions.

However, as educators it is not our task to only teach things that are fun. Engineering is not necessarily always fun. Besides, it is strange that engineers do not see the need to do things so formally. We need to listen much more carefully to what people object to and then try and see if we can cope with that. "Light-weight" formal method techniques may be an option for people to start with, but they may not be able to serve us well for real world problems. One may look to verify either a sub-class of properties with such "light-weight" techniques or look at a very small part of the system to verify other kinds of properties. Part of the reluctance in using the "heavy-weight" formal methods, perhaps, is that they force the user to make decisions right away. Light-weight methods allow some sort of ambiguity, alternately, some level of abstraction that would help in deferring some decision which need not be taken immediately. Light-weight models could, therefore, be used as a throw-away paper prototype before one moves to heavy weight models, to get some assurance upfront.