

Software Development

Arts and Crafts, or Math and Science?

Math!

Richard P. Gabriel
Sun Microsystems, Inc.

Mathematics is lovely, and being a mathematician is great fun. I have both a bachelor's and master's degree in mathematics, and came fairly close to getting a PhD in math, too. Mathematics is about axiomatically creating worlds and exploring them. The main activity is proposing and proving theorems. I estimate that during the 8 years I studied mathematics I proved between 5,000 and 10,000 theorems. At least one of them was a major proof—a theorem Dana Scott could not prove.

One way to think of software is that the requirements are like a proposed theorem, and the actual software is its proof. The theorem is of the form: There exists a program that satisfies these requirements; and the proof is constructive: The software is constructed (and shown to satisfy the requirements, usually through testing). This way we can look at constructing software as a mathematical process. There is one small problem with it: Constructing a proof is an art. Being able to prove theorems is something that one can be good or bad at—one gets better with practice, and one gets better the fastest by learning under a master. People can be talented at it, and the talent is only sometimes related to intelligence. Mathematics is a practice in which proving—trying to generate a proof—is a mechanism for fine-tuning proposed theorems: If a theorem seems like it should be true but can't be proved, then perhaps some of the concepts are not right or not formulated properly.

We think that math and science are somehow more disciplined or methodical than art and craft, but they are all human activities, and when examined closely they share more than they differ. Mathematics is invention, and physics has shown that inventions of the human mind can be useful. Hm, just like software.