

Software System Engineering: Was fehlt noch?

David Lorge Parnas

Abstract

The recognition that software had to be constructed in the disciplined and science-based way that other products are constructed began with the study of operating systems. The first operating systems were simple programs designed to replace the computer operator. Soon they took responsibility for other tasks such as permitting simultaneous execution of several user jobs at once and managing shared resources. They became the most complex programs in widespread use.

The problems that were first encountered in operating systems are now present in many other software products. Ideas that were pioneered in operating systems are now commonly used in those products. Thanks to advances in hardware and software, computers can perform services that were unimaginable when we started. However, many problems remain. Software products commonly have a number of “bugs” and other problems that we would not accept in a car or an elevator. Things essential in a mature profession are missing. This talk discusses three of them, viz:

- Education that prepares developers to apply science, education, and discipline to software tasks
- Rigid entrance standards for the profession
- Professional documentation standards similar to those used in other engineering disciplines

The least discussed of the three is documentation. The talk shows how we can use structured mathematical notation to provide precise documentation that is complete and useful to developers, reviewers, and maintainers.