

Laws and Principles of Evolution

Dewayne E. Perry
Electrical and Computer Engineering
The University of Texas at Austin
ECE Dept, Mail Stop C0803
Austin TX 78712-1084
perry@ece.utexas.edu
www.ece.utexas.edu/~perry/

First, we note that *Evolution* is one of Brooks' [1] essential characteristics of software systems: the only systems that are not evolving are the dead ones. Evolution is a basic fact of software life. The fact that we have evolution on multiple levels is often overlooked in considering the fact of evolution: local versus global, component versus system, internally versus externally motivated, etc.

Second, we note that while we have masses of data about local and component level evolution buried in our change and version management systems, we have done relatively little with that data to determine software evolution principles or theories for the underpinnings of software engineering. Principles are foundational in providing guidance in the various levels of evolution that take place – they form the bedrock of the software engineering enterprise.

Third, we note that we have very little data about software systems evolution - global evolution, evolution on a large scale. For our work on FEAST [2] we have had what amounts to a wealth of data: a mere handful of systems. The utility of these few sets of data is hampered further by the fact that there are at most about 25 data points (ie, 25 systems evolution instances) for each system.

Fourth, we note that we have in this small amount of data only a few of the important attributes needed to understand evolution deeply - namely, we have basic attributes such as system size, release dates (in some cases), etc. To substantiate laws of evolution we need both more systems and instances as well as more attributes for those systems and instances.

Fifth, we note that we need more than just (more) product data. We need process and organizational data to determine the fate of the FEAST hypothesis [3]. We need process and organizational data [4] in order to gain a deep understanding of the organizational processes and structures within which the systems are evolved. Determining the underlying correlations and causal mechanisms that show how feedback control works

within these social systems to effect the evolution within the software systems requires broader data that we currently have or currently are considering.

The understanding of system evolution is of fundamental importance to the software engineering enterprise. Lehman's Laws have been a fundamental and seminal start on the road to that understanding. However, we have a long way to go. We need more instances of systems and their evolution. We need more attributes of those evolutions. We need related process and organizational data. It is a research area that is a rich field that will yield a deep understanding of the phenomena of system evolution. But, we have a long way to go and a lot of work to do.

References

- [1] F. P. Brooks Jr, No Silver Bullet, in *The Mythical Man Month: Essays on Software Engineering*, Anniversary Edition, Addison Wesley, 1995.
- [2] M. M. Lehman, D. E. Perry and J. F. Ramil. Implications of Evolution Metrics on Software Maintenance. *Proc. ICSM'98*, November 1998.
- [3] M. M. Lehman, D. E. Perry and W. M. Turski, Why is it so hard to find Feedback Control in Software Processes?, *Proc. of the 19th Australasian Computer Science Conference*, Melbourne AUS, January 1996
- [4] N. Staudenmayer, T. L. Graves, J. S. Marron, A. Mockus, H. Siy, L. G. Votta, and D. E. Perry, Adapting to a new environment: how a legacy software organization copes with volatility and change," in *5th International Product Development Conference*, Como, Italy, 1998.