Joe Klemmer

Ms. Jill Coddington

SG150-OL-D2

Evolution of Software Cost and Complexity Since the 1960s

The evolution of software in cost and complexity relates directly to advances in hardware. As computers got smaller and more powerful, larger, highly intricate programs were being developed. The ratio of hardware cost to people cost started out with the vast majority on the former. Through the 60s and 70s, hardware costs fell rapidly while their power and speed shot through the roof. During this time, it became clear that some kind of methodology was needed to bring the exorbitant cost of software development under some control. As excerpt from an article in Dr. Dobb’s website, “As a result, it was more desirable and now finally economical to automate more and more applications of increasing complexity.”

Over time, programming languages advanced with the introduction of new software development paradigms. The 70’s saw the development and growth of Structured Programming. This design methodology worked well up to a point. However, ‘as Stein observes, “Structured programming appears to fall apart when applications exceed 100,000 lines or so of code.”’ In the 80s, Object-Oriented methodology became popular for its ability to compartmentalize data and the code to manipulate said data. Object-Oriented Design (OOD) allows for better organization of code and resources. Finally, in the late 90s and early into the 21st century, newer methods like Extreme Programming and Agile Programming came into practice along with Rapid Application Development languages.

In conclusion, with hardware becoming more powerful and their costs falling like a stone, it is the software costs, specifically in man-hours, that have become the “make or break” aspect of software development.

Works Cited

Agile Programming. Agile Programming Methodology. 2009. 31 May 2009 <http://agileprogramming.org/>.

Baker, F. T. "Structured programming in a production programming environment." 1975. Association for Computing Machinery. 31 May 2009 <http://portal.acm.org/citation.cfm?id=808437>.

Dr. Dobb's. Software Complexity: Bringing Order to Chaos. 4 June 2007. 31 May 2009 <http://www.ddj.com/architect/199901062?pgno=2>.

Grady Booch, Robert A. Maksimchuk, Michael W. Engel, Bobbi J. Young, Jim Conallen, Kelli A. Houston. Object-Oriented Analysis and Design with Applications, 3rd Edition. Addison-Wesley Professional, 2007.

Wells, Don. Extreme Programming: A gentle introduction. 17 February 2006. 31 May 2009 <http://www.extremeprogramming.org/>.