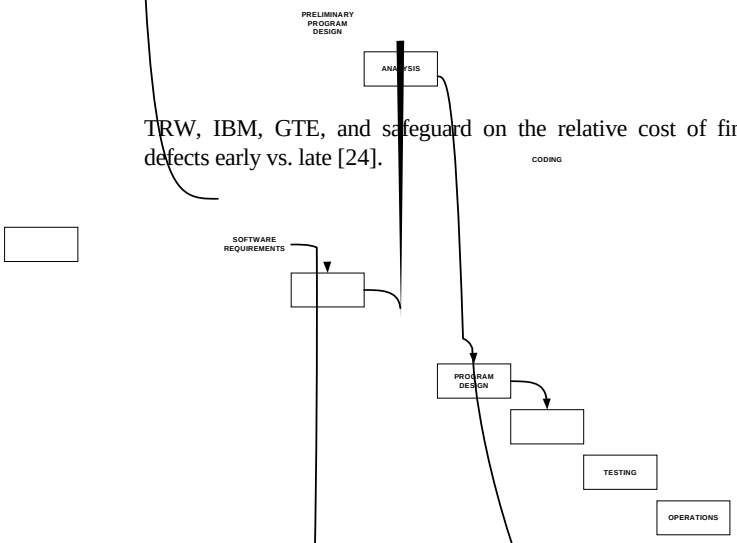
# A View of 20th and 21st Century Software Engineering

Barry Boehm
University of Southern California
University Park Campus, Los Angeles
boehm@cse.usc.edu

## ABSTRACT

George Santayana's statement, "Those who cannot remember the past are condemned to repeat it," is only half true. The past also includes successful histories. If you haven't been made aware of themo

**2.1**




TRW, IBM, GTE, and safeguard on the relative cost of finding defects early vs. late [24].

The rise in quantitati’e methods in the late 1970’s helped identify the major le’erage points for impro’ing software producti’ity. Distr 0on8.1(t) o17.8(t) ettrrt7t poihat6()11.2(ity)24.5( )13.4(3)err

automation, from the “essential” tasks unavoidably requiring syntheses of human expertise, judgment, and collaboration. The

adopted by Rational/IBM as the phase gates in the Rational Unified Process [87][133][84], and as such have been used on many

that not all these LOCS add value for their customers. But one could raise the same objections for all transistors being added to chips

groupware. Most software packages are oriented around individual use; just determining how best to support groups will take a good deal of research and experimentation.

**Software-Intensive Systems of Systems**

Concurrently between now and into the 2010's, the ability of organizations and their products, systems, and services to compete,

expert feedback on missing portions. It will enable simpler brute-force solutions such as exhaustive case evaluation vs. complex logic.

– Be skeptical about silver bullets, and one-size-fits-all solutions.

**From the 1990's**

[53] Cusumano, M., Selby, R.: Microsoft's Secrets. HarperCollins (1996)

[54] Dahl, O., and Nygaard, K. Simula: a language for programming and description of discrete event systems. Tech. rep., Norwegian Computing Center, 1967.

[55] DeMarco, T., Structured analysis and system specification. Prentice Hall, 1978.

[56] Dijkstra, E. *Cooperating Sequential Processes*. Academic Press, 1968

[57] Drexler, E.K.: Engines of Creation. Anchor Press (1986)

[58] Drexler, K.E., Peterson, C., Pergamit, G.: Unbounding the

[94] Littlewood, B. and Verrall, J. L., A Bayesian Reliability Growth Model for Computer Software, *Appl. Statist.*, Vol. 22, pp. 332-346 (1973)

[95] Maier, M., Architecting Principles for Systems of Systems,

[130] Redwine, S. and Riddle, W., Software Technology maturation, *Proceedings of the 8th ICSE*, 1985.

[131]