Software Maintenance Costs

(C) Jussi Koskinen
School of Computing, University of Eastern Finland, Joensuu, Finland
https://wiki.uef.fi/display/tktWiki/Jussi+Koskinen
April 30, 2015

Abstract
Software maintenance and evolution is a considerably understudied area while taking into account its cost effects. This document lists some interesting figures on proportional and absolute maintenance costs, proportions of the main task types, and amount and nature of the existing legacy code. These figures are based on empirical data. Although there has not been much empirical research on this particular area, the magnitude of the maintenance cost effects is clearly identifiable.

1. Proportional software maintenance costs

The relative cost for maintaining software and managing its evolution now represents more than 90% of its total cost. This is refered to as *legacy crisis* by Seacord *et al.* (2003). Various studies on this subject are described in **Table 1**.

Year	Proportion of software maintenance costs	Definition	Reference
2000	>90%	Software cost devoted to system maintenance & evolution / total software costs	Erlikh (2000)
1993	75%	Software maintenance / information system budget (in Fortune 1000 companies)	Eastwood (1993)
1990	>90%	Software cost devoted to system maintenance & evolution / total software costs	Moad (1990)
1990	60-70%	Software maintenance / total management information systems (MIS) operating budgets	Huff (1990)
1988	60-70%	Software maintenance / total management information systems (MIS) operating budgets	Port (1988)
1984	65-75%	Effort spent on software maintenance / total available software engineering effort.	McKee (1984)
1981	>50%	Staff time spent on maintenance / total time (in 487 organizations)	Lientz & Swanson (1981)
1979	67%	Maintenance costs / total software costs	Zelkowitz *et al.* (1979)

Table 1. Proportional software maintenance costs for its supplier.

2. Absolute software maintenance costs

- Annual software maintenance cost in USA has been estimated to be more than **$70 billion** (Sutherland, 1995; Edelstein, 1993).
- E.g. in USA, the federal government alone spent about **$8.38 billion** during a 5-year period to the Y2K-bug corrections.
- At company-level, e.g. Nokia Inc. used about **$90 million** for preventive Y2K-bug corrections.

3. Maintenance task types

- About **65%** of maintenance was found to be *perfective* by Lientz & Swanson (1981).
- About **75%** of maintenance costs are spent for providing *enhancements* (in the form of adaptive and perfective maintenance) (Martin, 1983; Nosek & Palvia, 1990; van Vliet, 2000).
- Studies of software maintainers have shown that approximately **50%** of their time is spent in the process of *understanding the code* that they are to maintain (Fjeldstad & Hamlen, 1983; Standish, 1984).

4. Legacy code amount

- In 1990 there were an estimated **120 billion lines** of source code being maintained (Ulrich, 1990).
- In 2000 there are already about **250 billion lines** of source code being maintained, and that number is increasing (Sommerville, 2000).
- An average Fortune 100 company maintains **35 million lines** of code (Müller *et al.*, 1994).
- These companies add in average **10%** each year only in enhancements (Müller *et al.*, 1994).
- As a result, the amount of code maintained doubles in size every **7 years** (Müller *et al.*, 1994).
- Older languages are not dead. E.g. **70%** or more of the still active business applications are written in COBOL (Giga Information Group).
- There are at least **200 billion lines** of COBOL-code still existing in mainframe computers alone (Gartner Group).

References

1) Eastwood, A. (1993). "Firm fires shots at legacy systems". *Computing Canada* **19** (2), p. 17.

2) Edelstein, D. (1993). "Report on the IEEE STD 1219 – 1993 – Standard for Software Maintenance". *ACM SIGSOFT Software Engineering Notes* **18** (4), p. 94.

3) Erlikh, L. (2000). "Leveraging legacy system dollars for E-business". *(IEEE) IT Pro*, May/June 2000, 17-23.

4) Fjeldstad, R. & Hamlen, W. (1983). "Application program maintenance-report to to our respondents". *Tutorial on Software Maintenance*, 13-27. Parikh, G. & Zvegintzov, N. (Eds.). IEEE Computer Soc. Press.

5) Huff, S. (1990). "Information systems maintenance". *The Business Quarterly* **55**, 30-32.

6) Lientz, B.P. & Swanson, E. (1980). "*Software Maintenance Management: A Study of the Maintenance of Computer Application Software in 487 Data Processing Organizations*". Addison-Wesley: Reading, MA, 214 p.

7) Lientz, B.P. & Swanson, E. (1981). "Problems in application software maintenance". *Communications of the ACM* **24** (11), 763-769.

8) Martin, J. (1983). "*Software Maintenance: The Problem and Its Solution*". Prentice Hall, 472 p.

9) McKee, J. (1984). "Maintenance as a function of design". *Proceedings of the AFIPS National Computer Conference*, 187-193.

10) Moad, J. (1990). "Maintaining the competitive edge". *Datamation* 61-62, 64, 66.

11) Müller, H., Wong, K. & Tilley, S. (1994). "Understanding software systems using reverse engineering technology". *The 62nd Congress of L'Association Canadienne Francaise pour l'Avancement des Sciences Proceedings (ACFAS)*, **26** (4), 41-48.

12) Nosek, J. & Palvia, P. (1990). "Software maintenance management: changes in the last decade". *Journal of Software Maintenance: Research and Practice* **2** (3), 157-174.

13) Port, O. (1988). "The software trap – automate or else". *Business Week* **3051** (9), 142-154.
14) Seacord, R., Plakosh, D. & Lewis, G. (2003). "*Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices*" (SEI Series in Software Engineering). Addison-Wesley.
15) Sommerville, I. (2000). "*Software Engineering (6th Edition)*". Addison-Wesley.
16) Standish, T. (1984). "An essay on software reuse". *IEEE Transactions on Software Engineering* **SE-10** (5), 494-497.
17) Sutherland, J. (1995). "Business objects in corporate information systems". *ACM Computing Surveys* **27** (2), 274-276.
18) Ulrich, W. (1990). "The evolutionary growth of software engineering and the decade ahead". *American Programmer* **3** (10), 12-20.
19) van Vliet, H. (2000). "*Software Engineering: Principles and Practice*". Wiley.
20) Zelkowitz, M., Shaw, A. & Gannon, J. (1979). "*Principles of Software Engineering and Design*". Prentice-Hall.

Updated: April 30, 2015 by Jussi Koskinen.