



注解与参考文献

第 1 章

1. Ershov 认为，编程是一种乐趣和苦恼共存的活动。A. P. Ershov, “Aesthetics and the human factor in programming,” *CACM*, 15,7(July,1972), pp. 501—505.

第 2 章

1. Bell 电话实验室的 V. A. Vyssotsky 估计一个大项目必须维持每年 30% 的人员投入。这导致巨大的压力，甚至限制了根本非正式结构和沟通的改进。第 7 章已对此进行了讨论。麻省理工学院的 F. J. Corbató 指出，一个长期的项目必须预见到每年有 20% 的人员更替，这些人必须进行技术上的培训，并整合到原有结构中。

2. 在 International Computers Limited 任职的 C. Portman 提出：“当所有的一切看上去可以工作，已经被集成时，你至少还有 4 个多月的工作需要完成。”有关其他的进度划分，请参见：Wolverton, R. W., “The cost of developing large-scale software,” *IEEE Trans. on Computers*, C-23, 6(June, 1974) pp.615—636。

3. 图 2-5、图 2-6、图 2-7、图 2-8 出自 Jerry Ogdin，他引用了本章的早期版本，并对相应的描述进行了改进。Ogdin, J. L., “The Mongolian

hordes versus superprogrammer,” *Info-systems*(Dec., 1972), pp.20—23.

第 3 章

1. Sackman, H., W. J. Erikson, and E. E. Grant, “Exploratory experimental studies comparing online and offline programming performance,” *CACM*, 11, 1(Jan., 1968), pp. 3—11.

2. Mills, H., “Chief programmer teams, principles, and procedures,” IBM Federal Systems Division Report FSC 715108, Gaithersburg, Md., 1971.

3. Baker, F. T., “Chief programmer team management of production programming,” *IBM Sys. J.* 11, 1 (1972).

第 4 章

1. Eschapasse, M., *Reims Cathedral*, Caisse Nationale des Monuments Historiques, Paris, 1967.

2. Brooks, F. P., “Architectural philosophy,” in W. Buchholz(ed.), *Planning A Computer System*. New York: McGraw-Hill, 1962.

3. Blaauw, G. A., “Hardware requirements for the fourth generation,” in F. Gruenberger (ed.), *Fourth Generation Computers*. Englewood Cliffs, N. J.: Prentice-Hall, 1970.

4. Brooks, F. P., and K. E. Iverson, *Automatic Data Processing, System/360 Edition*. New York: Wiley, 1969, Chapter 5.

5. Glegg, G. L., *The Design of Design*. Cambridge: Cambridge Univ. Press, 1969, 其在书中提出：“乍一看，用任何规则或者原理来约束创

造性思维的想法是一种阻碍，而不是一种帮助，但在实际情况下，却完全不是这样。规范的思维实际上促进而不是阻碍了灵感的产生。”

6. Conway, R. W., “The PL/C Compiler,” *Proceedings of a Conf. on Definition and Implementation of Universal Programming Languages*. Stuttgart, 1970.

7. 有关编程技术必要性的讨论，请参见：C. H. Reynolds, “What’s wrong with computer programming management?” in G. F. Weinwurm (ed.). *On the Management of Computer Programming*. Philadelphia: Auerbach, 1971, pp. 35—42。

第 5 章

1. Strachey, C., “Review of *Planning a Computer System*,” *Comp. J.*, 5, 2 (July, 1962), pp. 152—153.

2. 这仅仅适用于控制程序。OS/360 项目中的一些编译器开发团队正构建他们的第三个或第四个系统，他们卓越的产品展示了这一点。

3. Shell, D. L., “The Share 709 system: a cooperative effort” ; Greenwald, I. D., and M. Kane, “The Share 709 system: programming and modification” ; Boehm, E. M., and T. B. Steel, Jr. “The Share 709 system: machine implementation of symbolic programming” ; all in *JACM*, 6, 2(April, 1959), pp. 123—140.

第 6 章

1. Neustadt, R. E., *Presidential Power*. New York: Wiley, 1960, Chapter 2.

2. Backus, J. W., “The syntax and semantics of the proposed international algebraic language.” *Proc. Intl. Conf. Inf. Proc. UNESCO*, Paris, 1959, published by R. Oldenbourg, Munich, and Butterworth, London. Besides this, a whole collection of papers on the subject is contained in T. B. Steel, Jr. (ed.). *Formal Language Description Languages for Computer Programming*. Amsterdam: North Holland, (1966).

3. Lucas, P., and K. Walk, “On the formal description of PL/I,” *Annual Review in Automatic Programming Language*. New York: Wiley, 1962, Chapter 2, p. 2.

4. Iverson, K. E., *A Programming Language*. New York: Wiley, 1962, Chapter 2.

5. Falkoff, A. D., K. E. Iverson, E. H. Sussenguth, “A formal description of System/360,” *IBM Systems Journal*, 3, 3(1964), pp. 198—261.

6. Bell, C. G., and A. Newell, *Computer Structures*. New York: McGraw-Hill, 1970, pp. 120—136, 517—541.

7. Bell, C. G., private communication.

第 7 章

1. Parnas, D. L., “Information distribution aspects of design methodology,” Carnegie-Mellon Univ., Dept. of Computer Science Technical Report, February, 1971.

2. Copyright 1939, 1940 Street & Smith Publications, Copyright 1950, 1967 by Robert A. Heinlein. Published by arrangement with Spectrum Literary Agency.

第 8 章

1. Sackman, H., W. J. Erikson, and E. E. Grant, “Exploratory experimentation studies comparing online and offline programming performance,” *CACM*, 11, 1(Jan. 1968), 11, pp. 3—11.
2. Nanus, B., and L. Farr, “Some cost contributors to large-scale programs,” *AFIPS Proc. SJCC*, 25(Spring, 1964), pp. 239—248.
3. Weinwurm, G. F., “Research in the management of computer programming,” Report SP-2059, System Development Corp., Santa Monica, 1965.
4. Morin, L. H., “Estimation of resources for computer programming projects,” M. S. thesis. Univ. of North Carolina, Chapel Hill, 1974.
5. Portman, C., private communication.
6. E. F. Bardain 在 1964 年的一份未发表研究中指出，程序员实际的生产时间占 27%。参见：D. B. Mayer and A. W. Stalnaker, “Selection and evaluation of computer personnel, ” *Proc. 23rd ACM Conf.*, 1968, p. 661。
7. Aron, J. , private communication.
8. 该论文由小组分会提出，*AFIPS Proceedings* 并没有收录该论文。
9. Wolverton, R. W., “The cost of developing large-scale software,” *IEEE Trans. on Computers*, C-23, 6(June,1974), pp. 615—636. 这篇新近发表的重要文章包含的数据涉及了本章所讨论的许多问题，同时也核实了生产率方面的结论。
10. Corbató, F. J., “Sensitive issues in the design of multi-use systems.” 这是其于 1968 年在 Honeywell EDP 技术中心落成典礼上的公开演讲。

11. W. M. Taliaffero 在 Fortran 和 Cobol 编译器方面的生产率同样也达到了 2 400 语句/年。参见：“Modularity. The key to system growth potential,” *Software*, 1, 3 (July, 1971), pp. 245—257。

12. E. A. Nelson’s System Development Corp. Report TM-3225, *Management Handbook for the Estimation of Computer Programming Costs*, pp.66—67.该报告指出，尽管标准有较大的背离，仍然显示高级语言使生产率提高了 1~3 倍。

第 9 章

1. Brooks, F. P. and K. E. Iverson, *Automatic Data Processing, System/360 Edition*. New York: Wiley, 1969, Chapter 6.

2. Knuth, D. E., *The Art of Computer Programming*. Vols. 1—3. Reading, Mass.: Addison-Wesley, 1968, ff.

第 10 章

1. Conway, M. E., “How do committees invent?” *Datamation*, 14,4(April, 1968), pp. 28—31.

第 11 章

1. 1932 年 5 月 22 日在 Oglethorpe 大学的演讲。

2. 一本描述 Multics 在两个成功系统上所获得经验的书籍是：F. J. Corbató, J. H. Saltzer, and C. T. Clingen, “Multics—the first seven years,” *AFIPS Proc SJCC*, 40(1972), pp. 571—583。

3. Cosgrove, J., “Needed: a new planning framework,” *Datamation*, 17, 23(Dec., 1971), pp. 37—39.

4. 设计变更的问题是很复杂的，本文讨论的则比较简单，具体内容请参见：J. H. Saltzer, “Evolutionary design of complex systems,” in D. Eckman (ed.), *Systems: Research and Design*. New York: Wiley, 1961. 即使所有的事都已经提出和完成，我依然提倡构建一个计划好抛弃的先导系统。

5. 该现象在以下两篇文章中进行了讨论：Campbell, E., “Report to the AEC Computer Information Meeting,” December, 1970. J. L. Ogdin, “Designing reliable software,” *Datamation*, 18, 7(July, 1972), pp. 71—78. 至于曲线是否会再次下降，我的具有丰富经验的朋友们各执己见。

6. Lehman, M., and L. Belady, “Programming system dynamics,” given at the ACM SIGOPS Third Symposium on Operating System Principles, October, 1971.

7. Lewis, C. S., *Mere Christianity*. New York: Macmillan, 1960, p. 54.

第 12 章

1. 同时请参见：J. W. Pomeroy, “A guide to programming tools and techniques,” *IBM Sys. J.*, 11,3(1972), pp. 234—254.

2. Landy B., and R. M. Needham, “Software engineering techniques used in the development of the Cambridge Multiple-Access System,” *Software*, 1,2 (April, 1971), pp. 167—173.

3. Corbató F. J. , “PL/I as a tool for system programming,” *Datamation*, 15, 5(May, 1969), pp. 68—76.

4. Hopkins, M., “Problems of PL/I for system programming” IBM Research Report RC 3489, Yorktown Heights, N. Y., August 5, 1971.

5. Corbató, F. J., J. H. Saltzer, and C. T. Clingen, “Multic—the first seven years,” *AFIPS Proc SJCC*, 40(1972), pp. 571—582. “为了达到最优性能，仅有少数几个领域的 PL/I 程序重新用机器语言进行了改写。而大多最初使用机器语言编写的程序都用 PL/I 重新进行编写，以提高它们的可维护性。”

6. 引用了 Corbató 论文中的参考资料第 3 条：“PL/I is here now and the alternatives are still untested.” 同时请参见一些持有反面意见的文章：Henricksen, J. O. and R. E. Merwin, “Programming language efficiency in real-time software systems”, *AFIPS Proc SJCC*, 40(1972) pp. 155—161。

7. 并不是所有人都同意。在一次私下的交流中，Harlan Mills 指出：“我的经验开始告诉我，在产品开发中，被安排到终端面前的是秘书。其思想是使编程成为在众多团队成员监督下更加大众化的实践，而不是一项专有的技术。”

8. Harr, J., “Programming Experience for the Number 1 Electronic Switching System,” paper given at the 1969 SJCC.

第 13 章

1. Vyssotsky, V. A., 1972 年在美国 Chapel Hill, N. C. 举办的计算机程序测试方法讨论会(The Computer Program Test Methods Symposium)的演讲 “Common sense in designing testable software”。Vyssotsky 的大多数演讲都收录在：Hetzel, W. C. (ed.), *Program Test Methods*. Englewood Cliffs, N. J.: Prentice-Hall, 1972, pp. 41—47。

2. Wirth, N., “Program development by stepwise refinement,” *CACM* 14, 4(April, 1971), pp. 221—227. 同时请参见：Mills, H. “Top-down programming in large systems,” in R. Rustin (ed.). *Debugging Techniques in Large Systems*. Englewood Cliffs, N. J.: Prentice-Hall, 1971, pp. 41—55

and Baker F. T., “System quality through structured programming,” *AFIPS Proc FJCC*, 41-I(1972), pp. 339—343.

3. Dahl, O. J., E. W. Dijkstra, and C. A. R. Hoare, *Structured Programming*. London and New York: Academic Press, 1972. 该专栏包括了最完整的讨论。同时请参见 Dijkstra 早期的文章: “GOTO statement considered harmful,” *CACM*, 11,3(March, 1968), pp. 147—148。

4. Böhm, C., and A. Jacopini, “Flow diagrams, Turing machines, and languages with only two formation rules,” *CACM*, 9, 5(May, 1966), pp. 366—371.

5. Codd, E. F., E. S. Lowry, E. McDonough, and C. A. Scalzi, “Multiprogramming STRETCH: Feasibility considerations,” *CACM*, 2, 11(Nov., 1959), pp. 13—17.

6. Strachey, C., “Time sharing in large fast computers,” *Proc. Int. Conf. on Info. Processing*, UNESCO (June, 1959), pp. 336—341. 参见 Codd 在第 341 页上的评论, 他汇报了类似 Strachey 论文中所建议工作的进展状况。

7. Corbató, F. J., M. Merwin-Daggett, R. C. Daley, “An experimental time-sharing system,” *AFIPS Proc. SJCC*, 2, (1962), pp. 335—344. 重印于: S. Rosen, *Programming Systems and Languages*. New York: McGraw-Hill, 1967, pp. 683—698.

8. Gold, M. M., “A methodology for evaluating time-shared computer system usage,” Ph. D. dissertation, Carnegie-Mellon University, 1967, p. 100.

9. Gruenberger, F., “Program testing and validating,” *Datamation*, 14,7(July, 1968), pp. 39—47.

10. Ralston, A., *Introduction to Programming and Computer Science*. New York: McGraw-Hill, 1971, pp. 237—244.

11. Brooks, F. P., and K. E. Iverson, *Automatic Data Processing, System/360 Edition*. New York: Wiley, 1969, pp. 296—299.

12. F. M. Trapnell 提出了一种开发良好的规格说明、系统构建及测试处理的方法, 请参见: F.M.Trapnell, “A systematic approach to the development of system programs,” *AFIPS Proc SJCC*, 34 (1969) pp. 411—418.

13. 实时系统需要环境仿真器。有关举例请参见: M. G. Ginzberg, “Notes on testing real-time system programs,” *IBM Sys. J.*, 4, 1(1965), pp. 58—72.

14. Lehman, M., and L. Belady, “Programming system dynamics,” given at the ACM SIGOPS Third Symposium on Operating System Principles, October, 1971.

第 14 章

1. See C. H. Reynolds, “What’s wrong with computer programming management?” in G. F. Weinwurm (ed.), *On the Management of Computer Programming*. Philadelphia: Auerbach, 1971, pp. 35—42.

2. King, W. R., and T. A. Wilson, “Subjective time estimates in critical path planning—a preliminary analysis,” *Mgt. Sci.*, 13, 5(Jan., 1967), pp. 307—320, and sequel, W.R. King, D. M. Witterrongel, K. D. Hezel, “On the analysis of critical path time estimating behavior,” *Mgt. Sci.*, 14, 1(Sept., 1967), pp. 79—84.

3. 有关更详细的讨论请参见：Brooks, F. P., and K. E. Iverson, *Automatic Data Processing, System/360 Edition*, New York: Wiley, 1969, pp. 428—430。

4. Private communication.

第 15 章

1. Goldstine, H. H., and J. von Neumann, “Planning and coding problems for an electronic computing instrument,” Part II, Vol. 1., report prepared for U.S. Army Ordinance Department, 1947; reprinted in J. von Neumann, *Collected Works*, A. H. Taub (ed.), Vol. v., New York: McMillan, pp. 80—151.

2. Private communication, 1957. 该观点发表于：Iverson, K. E., “The use of APL in Teaching,” Yorktown, N.Y.: IBM Corp., 1969。

3. 有关 PL/I 的另外一个例子，请参见：A.B. Walter and M. Bohl, “From better to best—tips for good programming,” *Software Age*, 3, 11(Nov., 1969), pp. 46—50. 相同的技术可以使用在 Algol 中，Colorado 大学的 D. E. Lang 有一个称为 “STYLE” 的 Fortran 格式的程序，可以达到上述效果。同时请参见：D. D. McCracken and G. M. Weinberg, “How to write a readable FORTRAN program,” *Datamation*, 18, 10(Oct., 1972), pp. 73—77。

第 16 章

1. 标题为 “No Silver Bullet” 的论文出自：Information Processing 1986, the Proceedings of the IFIP Tenth World Computing Conference, edited by H. - J. Kugler (1986), pp. 1069—1076. 在获得 IFIP 和 Elsevier Science B. V., Amsterdam, The Netherlands 的许可后重印。

2. Parnas, D. L., “Designing software for ease of extension and contraction,” *IEEE Trans. on SE*, 5, 2 (March, 1979), pp. 128—138.

3. Booch, G., “Object-oriented design,” in *Software Engineering with Ada*. Menlo Park, Calif.: Benjamin/Cummings, 1983.

4. Mostow, J., ed., Special Issue on Artificial Intelligence and Software Engineering, *IEEE Trans. on SE*, 11, 11 (Nov., 1985).

5. Parnas, D. L., “Software aspects of strategic defense systems,” *Communications of the ACM*, 28, 12 (Dec., 1985), pp. 1326—1335. Also in *American Scientist*, 73,5 (Sept.—Oct., 1985), pp. 432—440.

6. Balzer, R., “A 15-year perspective on automatic programming,” in Mostow, op.cit.

7. Mostow, op.cit.

8. Parnas, 1985, op.cit.

9. Raeder, G., “A survey of current graphical programming techniques,” in R. B. Grafton and T. Ichikawa, eds., Special Issue on Visual Programming, *Computer*, 18, 8 (Aug., 1985), pp. 11—25.

10. 该主题在本书第 15 章讨论。

11. Mills, H.D., “Top-down programming in large systems,” *Debugging Techniques in Large Systems*, R. Rustin, ed., Englewood Cliffs, N. J., Prentice-Hall, 1971.

12. Boehm, B. W., “A spiral model of software development and enhancement,” *Computer*, 20, 5 (May, 1985), pp. 43—57.

第 17 章

引用但未被引证的材料来自私下的交流。

1. Brooks, F. P., “No silver bullet—essence and accidents of software engineering,” in *Information Processing 86*, H. J. Kugler, ed., Amsterdam: Elsevier Science (North Holland), 1986, pp. 1069—1076.

2. Brooks, F. P., “No silver bullet—essence and accidents of software engineering,” *Computer* 20, 4 (Apr., 1987), pp. 10—19.

3. 许多来往信件，刊登在 *Computer* 杂志，1987 年 7 月。

“非常高兴地看到《没有银弹》没有得到任何大奖”——Bruce M. Skwiersky 的评论成为 *Computer Reviews* 1988 年选出的最佳评论。同时宣布上述奖项并重提了 Skwiersky 观点的还有：E.A. Weiss, “Editorial,” *Computing Reviews* (June, 1989), pp. 283—284。不过该评论有一个重大的失误：“sixfold”应该为“106”。

4. “根据亚里士多德的经院哲学中的观点，次要(accident)是不属于事物必要或者根本的属性，而是作为其他原因引起的后果。” *Webster's New International Dictionary of the English Language*, 2d ed., Springfield, Mass.: G. C. Merriam, 1960.

5. Sayers, Dorothy L., *The Mind of the Maker*. New York: Harcourt, Brace, 1941.

6. Glass, R. L., and S. A. Conger, “Research software tasks: Intellectual or clerical?” *Information and Management*, 23, 4 (1992). 作者提出了关于软件需求的度量结果是 80% 的智力和 20% 的文书工作。关于应用软件维护，Fjelstadt 和 Hamlen 在 1979 年得到了相同的结果。对于完整的任务而言，据我所知还没有类似的测量。

7. Herzberg, F., B. Mausner, and B. B. Sayderman. *The Motivation to Work*, 2nd ed. London: Wiley, 1959.

8. Cox, B. J., "There is a silver bullet," *Byte* (Oct., 1990), pp. 209—218.

9. Harel, D., "Biting the silver bullet: Toward a brighter future for system development," *Computer* (Jan., 1992), pp. 8—20.

10. Parnas, D. L., "Software aspects of strategic defense systems," *Communications of the ACM*, 28, 12 (Dec., 1985), pp. 1326—1335.

11. Turski, W. M., "And no philosophers' stone, either," in *Information Processing 86*, H. J. Kugler, ed., Amsterdam: Elsevier Science(North Holland),1986, pp. 1077—1080.

12. Glass, R. L., and S. A. Conger, "Research Software Tasks: Intellectual or Clerical?" *Information and Management*, 23, 4 (1992), pp. 183—192.

13. *Review of Electronic Digital Computers, Proceedings of a Joint AIEE-IRE Computer Conference*(Philadelphia, Dec. 10—12, 1951). New York: American Institute of Electrical Engineers, pp. 13—20.

14. *Ibid.*, pp. 36, 68, 71, 97.

15. *Proceedings of the Eastern Joint Computer Conference*, (Washington, Dec. 8—10, 1953). New York: Institute of Electrical Engineers, pp. 45—47.

16. *Proceedings of the 1955 Western Joint Computer Conference*, (Los Angeles, March 1—3, 1955). New York: Institute of Electrical Engineers.

17. Everett, R. R., C. A. Zraket, and H. D. Benning-ton, “SAGE—A data processing system for air defense,” *Proceedings of the Eastern Joint Computer Conference*, (Washington, Dec. 11—13, 1957). New York: Institute of Electrical Engineers.

18. Harel, D., H. Lachover, A. Naamad, A. Pnueli, M. Politi, R. Sherman, A. Shtul-Trauring, “Statemate: A working environment for the development of complex reactive systems,” *IEEE Trans. on SE*, 16, 4 (1990), pp. 403—444.

19. Jones, C., *Assessment and Control of Software Risks*. Englewood Cliffs, N. J.: Prentice-Hall, 1994. p. 619.

20. Coqui, H., “Corporate survival: The software dimension,” *Focus’89*, Cannes, 1989.

21. Coggins, James M., “Designing C++ libraries,” *C++Journal*, 1, 1 (June, 1990), pp. 25—32.

22. 时态是将来时，我所了解到的是，没有类似关于应用了 15 次的报告。

23. Jones, op.cit., p. 604.

24. Huang, Weigiao, “Industrializing software production,” *Proceedings ACM 1988 Computer Science Conference*, Atlanta, 1988. 我觉得在这样的安排中，缺乏个人工作机会的增长。

25. 1994 年 9 月的 *IEEE Software* 全篇讨论的都是有关这些的。

26. Jones, op.cit., p. 323.

27. Jones, op.cit., p. 329.

28. Yourdon, E., *Decline and Fall of the American Programmer*. Englewood Cliffs, N. J.: Yourdon Press, 1992, p. 221.

29. Glass, R. L., “Glass” (column), *System Development*, (Jan., 1988), pp. 4—5.

第 18 章

1. Boehm, B. W., *Software Engineering Economics*, Englewood Cliffs, N. J.: Prentice-Hall, 1981, pp. 81—84.

2. McCarthy, J., “21 Rules for Delivering Great Software on Time,” *Software World USA Conference*, Washington (Sept., 1994).

第 19 章

引用但未被引证的材料来自私下的交流。

1. 关于这个痛苦的话题，同时请参见：Niklaus Wirth “A plea for lean software,” *Computer*, 28, 2 (Feb., 1995), pp. 64—68。

2. Coleman, D., 1994, “Word 6.0 packs in features; update slowed by baggage,” *MacWeek*, 8, 38 (Sept. 26, 1994), p. 1.

3. 在发布安装之后，发表了一些机器语言和编程语言命令的概率数据。有关举例可参见：J. Hennessy and D. Patterson, *Computer Architecture*. 尽管这些概率数据从不会精确匹配，但对构建后续的产品非常有用。据我所知，在产品的设计之前没有任何书面的概率估计，将**事先估计**和**实际情况**进行比较就更少了。Ken Brooks 建议，Internet 上的公告牌现在提供了一种成本更为低廉的从新产品的预期用户那里获取数据的方法，即使

只有少数人会对此做出答复。

4. Conklin, J., and M. Begeman, “gIBIS: A Hypertext Tool for Exploratory Policy Discussion,” *ACM Transactions on Office Information Systems*, Oct. 1988, pp. 303—331.

5. Englebart, D., and W. English, “A research center for augmenting human intellect,” *AFIPS Conference Proceedings, Fall Joint Computer Conference*, San Francisco (Dec. 9—11, 1968), pp. 395—410.

6. Apple Computer, Inc., *Macintosh Human Interface Guidelines*, Reading, Mass.: Addison-Wesley, 1992.

7. 尽管 Apple Desk Top Bus 可以从电气原理上控制两个鼠标，但操作系统并未提供类似功能。

8. Royce, W. W., 1970. “Managing the development of large software systems: Concepts and techniques,” *Proceedings, WESCON*(Aug., 1970), reprinted in the *ICSE 9 Proceedings*. Royce 和其他人都认为软件过程自始至终不修订前期文档是不可能的；模型是作为理想情况和概念提出的。D. L. Parnas and P. C. Clements, “A rational design process: How and why to fake it,” *IEEE Transactions on Software Engineering*, SE-12,2(Feb., 1986), pp. 251—257.

9. DOD-STD-2167 重新制订的工作产生了 DOD-STD-2167A (1988)，它允许但并未制订新的模型如螺旋模型等。Boehm 的报告指出：不幸的是，2167A 所参考的军标 MILSPECS 和说明性的例子依然是面向瀑布模型的，因此瀑布模型依然继续被采用。Larry Druffel 和 George Heilmeyer 所领导的国防科学委员会(Defense Science Board Task Force)，在他们 1994 年的报告“Report of the DSB task force on acquiring defense software commercially”中曾提倡大规模使用更现代的模式。

10. Mills, H.D., “Top-down programming in large systems,” in *Debugging Techniques in Large Systems*, R.Rustin, ed. Englewood Cliffs, N. J.: Prentice-Hall, 1971.

11. Parnas, D. L., “On the design and development of program families,” *IEEE Trans. on Software Engineering*, SE-2, 1 (March, 1976), pp. 1—9; Parnas, D. L., “Designing software for ease of extension and contraction,” *IEEE Trans. on Software Engineering*, SE-5, 2 (March, 1979), pp. 128—138.

12. D. Harel, “Biting the silver bullet,” *Computer* (Jan., 1992), pp. 8—20.

13. 在信息隐藏方面，开创性论文包括：Parnas, D. L., “Information distribution aspects of design methodology,” Carnegie-Mellon, Dept. of Computer Science, Technical Report (Feb., 1971); Parnas, D. L., “A technique for software module specification with examples,” *Comm. ACM*, 5, 5 (May, 1972), pp. 330—336; Parnas, D. L. (1972). “On the criteria to be used in decomposing systems into modules,” *Comm. ACM*, 5, 12 (Dec., 1972), pp. 1053—1058.

14. “对象”的思想首先由 Hoare and Dijkstra 提出，但是第一个和最有影响力的案例是 Dahl and Nygaard 发明的 Simula-67 语言。

15. Boehm, B. W., *Software Engineering Economics*, Englewood Cliffs, N. J.: Prentice-Hall, 1981, pp. 83—94; 470—472.

16. Abdel-Hamid, T., and S. Madnick, *Software Project Dynamics: An Integrated Approach*, ch. 19, “Model enhancement and Brooks’s law.” Englewood Cliffs, N. J.: Prentice-Hall, 1991.

17. Stutzke, R. D., “A Mathematical Expression of Brooks’ s Law,”
In *Ninth International Forum on COCOMO and Cost Modeling*. Los
Angeles: 1994.

18. DeMarco, T., and T. Lister, *Peopleware: Productive Projects and
Teams*. New York: Dorset House, 1987.

19. Pius XI, Encyclical *Quadragesimo Anno*, [Ihm, Claudia Carlen,
ed., *The Papal Encyclicals 1903—1939*. Raleigh, N. C.: McGrath, p. 428.]

20. Schumacher, E. F., *Small Is Beautiful: Economics as if People
Mattered*, Perennial Library Edition. New York: Harper and Row, 1973, p.
244.

21. Schumacher, op.cit., p. 34.

22. 一则发人深省的海报声称：“言论自由属于拥有它们的人。”

23. Bush, V., “That we may think,” *Atlantic Monthly*, 176, 1 (Apr.,
1945), pp. 101—108.

24. Unix 的发明人，Bell Labs 的 Ken Thompson 很早就认识到大屏
幕对编程的重要性。他在原始的 Tektronix 电子显像管上发明了 120 行
代码的显示方法。他在整个高速显像管和小型窗口的时代中坚持使用该
终端。