

已经众所周知的是，顶尖的程序员比拙劣的程序员生产率要高出10倍（Sackman等人，1968）。麻烦在于，当你需要200名程序员时，找到200名顶尖的程序员非常困难，对于程序员的质量你不得不有所将就。

在任何大型的设计项目（软件或其他）中，同样重要的是需要体系结构的一致性。应该有一名才智超群的人对设计进行控制。Brooks引证兰斯大教堂^①作为大型项目的例子，兰斯大教堂的建造花费了几十年的时间，在这一过程中，后来的建筑师完全服从于完成最初风格的建筑师的规划。结果是其他欧洲大教堂无可比拟的建筑物的一致性。

在20世纪70年代，Harlan Mills把“一些程序员比其他程序员要好很多”的观察结果与对体系结构一致性的需要相结合，提出了首席程序员团队（chief programmer team）的范式（Baker, 1972）。他的思想是要像一个外科手术团队，而不是像一个杀猪屠夫团队那样组织一个程序员团队。不是每个人像疯子一样乱砍一气，而是由一个人掌握着手术刀，其他人在那里提供支持。对于一个10名人员的项目，Mills建议的团队结构如图13-10所示。

头 衔	职 责
首席程序员	执行体系结构设计并编写代码
副手	辅助首席程序员并为其提供咨询
行政主管	管理人员、预算、空间、设备、报告等
编辑	编辑文档，而文档必须由首席程序员编写
秘书	行政主管和编辑各需要一名秘书
程序文书	维护代码和文档档案
工具师	提供首席程序员需要的任何工具
测试员	测试首席程序员的代码
语言律师	兼职人员，他可以就语言向首席程序员提供建议

图13-10 Mills建议的10人首席程序员团队的分工

自从提出这一建议并付诸实施，30年过去了。一些事情已经变化（例如需要一个语言层——C比PL/I更为简单），但是只需要一名才智超群的人员对设计进行控制仍然是正确的。并且这名才智超群者在设计和编程上应该能够100%地起作用，因此需要支持人员。尽管借助于计算机的帮助，现在一个更小的支持人员队伍就足够了。但是在本质上，这一思想仍然是有效的。

任何大型项目都需要组织成层次结构。底层是许多小的团队，每个团队由首席程序员领导。在下一层，必须由一名经理人对一组团队进行协调。经验表明，你所管理的每一个人将花费你10%的时间，所以每10个团队的一个小组就需要一名全职的经理。这些经理人也必须被管理。

Brooks观察到，坏消息不能很好地沿着树向上传播。麻省理工学院的Jerry Saltzer将这一效应称为坏消息二极管（bad-news diode）。因为存在着在两千年前将带来坏信息的信使斩首的古老传统，所以首席程序员或经理人都不会愿意告诉他的老板项目延期了4个月，并且无论如何都没有满足最终时限的机会。因此，顶层管理者就项目的状态通常不明就里。当不能满足最终时限的情况变得十分明显时，顶层管理者的响应是增加人员，此时Brooks定律就起作用了。

实际上，大型公司拥有生产软件的丰富经验并且知道如果它随意地生产会发生什么，这样的公司趋向于至少是试图正确地做事情。相反，较小的、较新的公司，匆匆忙忙地希望其产品早日上市，不能总是仔细地生产他们的软件。这经常导致远远不是最优化的结果。

Brooks和Mills都没有预见到开放源码运动的成长。尽管该运动取得了某些成功，但是一旦新鲜感消失，它是否还是生产大量高质量软件的切实可行的模型还有待观察。回想早年无线电广播是由业余无线电操作人员占据支配地位的，但是很快就让位于商业无线电台和后来的商业电视台。值得注意的是，最为成功的开放源码软件项目显然使用了首席程序员模型，有一名才智超群者控制着体系结构设计（例如，Linus Torvalds控制着Linux内核，而Richard Stallman控制着GNU C编译器）。

① 兰斯（Reims）——法国东北部城市。——译者注