

人月神话[转 经典老文]

写在前面

本文是一篇写于40多年前的经典老文，原文是《人月神话》这本书中的一篇论文，作者Brooks。

相信不少程序员的书架上都有《人月神话》这本书。是否每个人都认真阅读了这本书，我深表怀疑。40多年过去了，本文中提到的错误仍在频繁复现：向进度落后的项目中增加人手，只会使进度更加落后。

美酒的酿造需要年头，美食的烹调需要时间；片刻等待，更多美味，更多享受。

□- 新奥尔良餐厅的菜单

□□- □

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">在众多软件项目中，缺乏合理的时间进度是造成项目滞后的最主要原因，它比其他所有因素加起来的影响还大。导致这种普遍性灾难的原因是什么呢？

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">首先，我们对估算技术缺乏有效的研究，更加严肃地说，它反映了一种悄无声息，但并不真实的假设——一切都将运作良好。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">第二，我们采用的估算技术隐含地假设人和月可以互换，错误地将进度与工作量相互混淆。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">第三，由于对自己的估算缺乏信心，软件经理通常不会有耐心持续地进行估算这项工作。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">第四，对进度缺少跟踪和监督。其他工程领域中，经过验证的跟踪技术和常规监督程序，在软件工程中常常被认为是无谓的举动。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">第五，当意识到进度的偏移时，下意识（以及传统）的反

应是增加人力。这就像使用汽油灭火一样，只会使事情更糟。越来越大的火势需要更多的汽油，从而进入了一场注定会导致灾难的循环。

进度监督是另一篇论文的主题，而本文中我们将对问题的其他方面进行更详细的讨论。

乐观主义

所有的编程人员都是乐观主义者。可能是这种现代魔术特别吸引那些相信美满结局的人；也可能是成百上千琐碎的挫折赶走了大多数人，只剩下了那些习惯上只关注结果的人；还可能仅仅因为计算机还很年轻，程序员更加年轻，而年轻人总是些乐观主义者——无论是什么样的程序，结果是毋庸置疑的：“这次它肯定会运行。”或者“我刚刚找出了最后一个错误。”

所以系统编程的进度安排背后的第一个假设是：

一切都将运作良好

，

每一项任务仅花费它所“应该”花费的时间

。

对这种弥漫在编程人员中的乐观主义，理应受到慎重的分析。

Dorothy Sayers

在她的 “

The Mind of the Maker

”一书中，将创造性活动分为三个阶段：

构思、实现和交流。书籍、计算机、或者程序的出现，首先是作为一个构思或模型出现在作者的脑海中，它与时间和空间无关。接着，借助钢笔、墨水和纸，或者电线、硅片和铁氧体，在现实的时间和空间中实现它们。然后，当某人阅读书本、使用计算机和运行程序的时候，他与作者的思想相互沟通，从而创作过程得以结束。

以上

Sayers

的阐述不仅仅可以描绘人类的创造性活动，而且类似于“基督的教义”，能指导我们的日常工作。对于创造者，只有在实现的过程中，才能发现我们构思的不完整

。

。

。

性和不一致性。因此，对于理论家而言，书写、试验以及“工作实现”是非常基本和必要的。

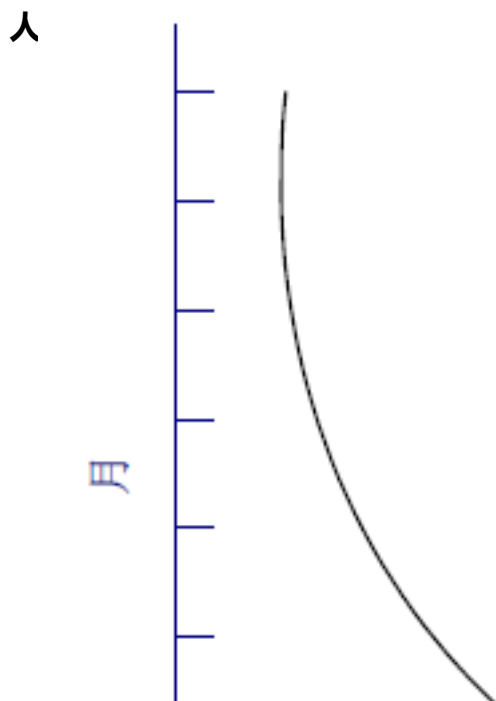
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:0pt">在许多创造性活动中，往往很难掌握活动实施的介质，例如木头切割、油漆、电器安装等。这些介质的物理约束限制了思路的表达，它们同样对实现造成了许多预料之外的困难。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:0pt">由于物理介质和思路中隐含的不完善性，实际实现起来需要花费大量的时间和汗水。对遇到的大部分实现上的困难，我们总是倾向于去责怪那些物理介质，因为它们不顺应“我们”设定的思路。其实，这只不过是我们的骄傲使判断带上了主观主义色彩。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:0pt">然而，计算机编程基于十分容易掌握的介质，编程人员通过非常纯粹的思维活动——概念以及灵活的表现形式来开发程序。正由于介质的易于驾驭，我们期待在实现过程中不会碰到困难，因此造成了乐观主义的弥漫。而我们的构思是有缺陷的，因此总会有

mso-bidi-font-family:Arial;color:black;mso-font-kerining:0pt">bug
宋体;color:black;mso-font-kerining:0pt">。也就是说，我们的乐观主义并不应该是理所应当的。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:0pt">在单个的任务中，“一切都将运转正常”的假设在时间进度上具有可实现性。因为所遇的延迟是一个概率分布曲线，“不会延迟”仅具有有限的概率，所以现实情况可能会像计划安排的那样顺利。然而大型的编程工作，或多或少包含了很多任务，某些任务间还具有前后的次序，从而一切正常的概率变得非常小，甚至接近于无。



10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">第二个谬误的思考方式是在估计和进度安排中使用的工作量单位：人月。成本的确随开发产品的人数和时间的不同，有着很大的变化，进度却不是如此。因此我认为

宋体;color:black;mso-font-kerning:0pt">用人月作为衡量一项工作的规模是一个危险和带有欺骗性的神话。

宋体;color:black;mso-font-kerning:0pt">它暗示着人员数量和时间是可以相互替换的。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">人数和时间的互换仅仅适用于以下情况：某个任务可以分解给参与人员，并且

宋体;color:black;mso-font-kerning:0pt">他们之间不需要相互的交流（图

Arial;color:black;mso-font-kerning:0pt">1

font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">)。

0pt">)。

mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">这在割小麦或收获棉花的工作中是可行的；而在系统编程中近乎不可能。

图1 人员和时间之间的关系——完全可以分解的任务

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">当任务由于次序上的限制不能分解时，人手的添加对进度没有帮助（图

Arial;color:black;mso-font-kerning:0pt">2

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">)。无论多少个母亲，孕育一个生命都需要十个月。由于调试、测试的次序特性，许多软件都具有这种特征。

图2 人员和时间之间的关系——无法分解的任务

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">对于可以分解，但子任务之间需要相互沟通和交流的任务，必须在计划工作中考虑沟通的工作量。因此，相同人月的前提下，采用增加人手来减少时间得到的最好情况，也比未调整前要差一些（图

mso-bidi-font-family:Arial;color:black;mso-font-kerning:0pt">3

宋体;color:black;mso-font-kerining:0pt" >) 。

图3 人员和时间之间的关系——需要沟通的可分解任务

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:0pt" > 沟通所增加的负担由两个部分组成，培训和相互的交流。每个成员需要进行技术、项目目标以及总体策略上的培训。这种培训不能分解，因此这部分增加的工作量随人员的数量呈线性变化

Arial;color:black;mso-font-kerining:0pt" > 1

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:0pt" > 。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:0pt" > 相互之间交流的情况更糟一些。如果任务的每个部分必须分别和其他部分单独协作，则工作量按照

mso-bidi-font-family:Arial;color:black;mso-font-kerining:0pt" > $n(n-1)/2$

宋体;color:black;mso-font-kerining:0pt" > 递增。一对一交流的情况下，三个人的工作量是两个人的三倍，四个人则是两个人的六倍。而对于需要在三四个人之间召开会议、进行协商、一同解决的问题，情况会更加恶劣。所增加的用于沟通的工作量可能会完全抵消对原有任务分解所产生的作用，此时我们会被带到图

mso-bidi-font-family:Arial;color:black;mso-font-kerining:0pt" > 4

宋体;color:black;mso-font-kerining:0pt" > 的境地。

图4 人员和时间之间的关系——关系错综复杂的任务

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:0pt" > 因为软件开发本质上是一项系统工作——错综复杂关系下的一种实践——沟通、交流的工作量非常大，它很快会消耗任务分解所节省下来的个人时间。从而，添加更多的人手，实际上是延长了，而不是缩短了时间进度。

系统测试

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:0pt" > 在时间进度中，顺序限制所造成的影响，没有哪个部分比单元调试和系统测试所受到的牵涉更彻底。而且，要求的时间依赖于所遇到的错误、缺陷数量以及捕捉它们的程度。理论上，缺陷的数量应该为零。但是，由于我们的乐观主义，通常实际出现的缺陷数量比预料的要多得多。因此，系统测试进度的安排常常是编程中最不合理的部分。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:0pt" > 对于软件任务的进度安排，以下是我使用了很多年的经验

法则：

mso-bidi-font-family:Arial;color:black;mso-font-kerning:0pt">1/3
宋体;color:black;mso-font-kerning:0pt">计划

mso-bidi-font-family:Arial;color:black;mso-font-kerning:0pt">1/6
宋体;color:black;mso-font-kerning:0pt">编码

mso-bidi-font-family:Arial;color:black;mso-font-kerning:0pt">1/4
宋体;color:black;mso-font-kerning:0pt">构件测试和早期系统测试

mso-bidi-font-family:Arial;color:black;mso-font-kerning:0pt">1/4
宋体;color:black;mso-font-kerning:0pt">系统测试，所有的构件已完成

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">在许多重要的方面，它与传统的进度安排方法不同：

Arial;color:black;mso-font-kerning:0pt">1.

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">分配给计划的时间比寻常的多。即便如此，仍不足以产生
详细和稳定的计划规格说明，也不足以容纳对全新技术的研究和摸索。

Arial;color:black;mso-font-kerning:0pt">2.

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">对所完成代码的调试和测试，投入近一半的时间，比平常
的安排多很多。

Arial;color:black;mso-font-kerning:0pt">3.

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">容易估计的部分，即编码，仅仅分配了六分之一的时间。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">通过对传统项目进度安排的研究，我发现很少项目允许为
测试分配一半的时间，但大多数项目的测试实际上是花费了进度中一半的时间。它们
中的许多项目，在系统测试之前还能保持进度。或者说，除了系统测试，进度基本能
保证

Arial;color:black;mso-font-kerning:0pt">2

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;

mso-font-kerning:0pt">特别需要指出的是，不为系统测试安排足够的时间简直就是一场灾难。因为延迟发生在项目快完成的时候。直到项目的发布日期，才有人发现进度上的问题。因此，坏消息没有任何预兆，很晚才出现在客户和项目经理面前。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">另外，此时此刻的延迟具有不寻常的、严重的财务和心理上的反应。在此之前，项目已经配置了充足的人员，每天的人力成本也已经达到了最大的限度。更重要的是，当软件用来支持其他的商业活动（计算机硬件到货，新设备、服务上线等等）时，这些活动延误出现即将发布前，那么将付出相当高的商业代价。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">实际上，上述的二次成本远远高于其他开销。因此，在早期进度策划时，允许充分的系统测试时间是非常重要的。

空泛的估算

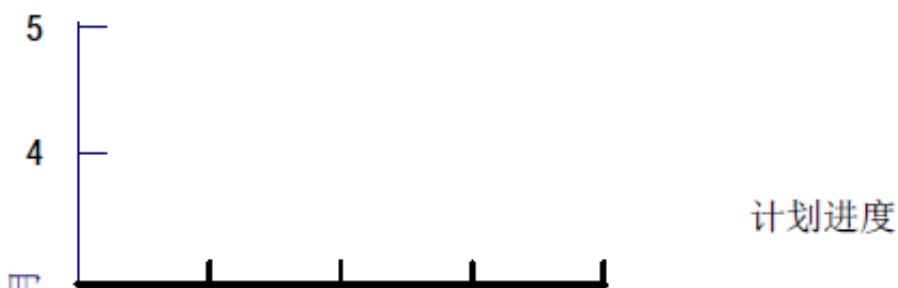
观察一下编程人员，你可能会发现，同厨师一样，某项任务的计划进度，可能受限于顾客要求的紧迫程度，但紧迫程度无法控制实际的完成情况。就像约好在两分钟内完成一个煎蛋，看上去可能进行得非常好。但当它无法在两分钟内完成时，顾客只能选择等待或者生吃煎蛋。软件顾客的情况类似。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">厨师还有其他的选择：他可以把火开大，不过结果常常是无法“挽救”的煎蛋——一面已经焦了，而另一面还是生的。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">现在，我并不认为软件经理内在的勇气和坚持不如厨师，或者不如其他工程经理。但为了满足顾客期望的日期而造成的不合理进度安排，在软件领域中却比其他的任何工程领域要普遍得多。而且，非阶段化方法的采用，少得可怜的数据支持，加上完全借助软件经理的直觉，这样的方式很难生产出健壮可靠和规避风险的估计。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">显然我们需要两种解决方案。开发并推行生产率图表、缺陷率、估算规则等等，而整个组织最终会从这些数据的共享上获益。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">或者，在基于可靠基础的估算出现之前，项目经理需要挺直腰杆，坚持他们的估计，确信自己的经验和直觉总比从期望派生出的结果要强得多。



10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">当一个软件项目落后于进度时，通常的做法是什么呢？自然是加派人手。如图
mso-bidi-font-family:Arial;color:black;mso-font-kerning:0pt">1
宋体;color:black;mso-font-kerning:0pt">至
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">4
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:
0pt">所示，这可能有所帮助，也可能无法解决问题。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">我们来考虑一个例子
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">3
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:
0pt">。设想一个估计需要
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">12
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:
0pt">个人月的任务，分派给
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">3
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:
0pt">个成员
"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;mso-font-kerning:0pt">4
宋体;color:black;mso-font-kerning:0pt">个月时间，在每个月的末尾安排了可测量的里程碑
mso-bidi-font-family:Arial;color:black;mso-font-kerning:0pt">A
宋体;color:black;mso-font-kerning:0pt">、
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;

mso-font-kerning:0pt">B
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">、
"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;mso-font-kerning:0pt">C
宋体;color:black;mso-font-kerning:0pt">、
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">D
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">(图
"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;mso-font-kerning:0pt">5
宋体;color:black;mso-font-kerning:0pt">)。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">现在假定两个月之后，第一个里程碑没有达到(图
Arial;color:black;mso-font-kerning:0pt">6
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">)。项目经理面对的选择方案有哪些呢？

Arial;color:black;mso-font-kerning:0pt">1.
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">假设任务必须按时完成。假设仅仅是任务的第一个部分估计不得当，即如图
Arial;color:black;mso-font-kerning:0pt">6
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">所示，则剩余了
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">9
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:0pt">个人月的工作量，时间还有两个月，即需要
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;

mso-font-kerning:0pt">4.5
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;ms
o-font-kerning:
0pt">个开发人员，所以需要在原来
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">3
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;ms
o-font-kerning:
0pt">个人的基础上增加
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">2
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;ms
o-font-kerning:
0pt">个人。

Arial;color:black;mso-font-kerning:0pt">2.
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:bl
ack;
mso-font-kerning:0pt">假设任务必须按时完成。假设整个任务的估计偏低，即如
图

Arial;color:black;mso-font-kerning:0pt">7
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:bl
ack;
mso-font-kerning:0pt">所示，那么还有
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:bl
ack;

mso-font-kerning:0pt">18
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;ms
o-font-kerning:
0pt">个人月的工作量以及
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">2
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;ms
o-font-kerning:
0pt">个月的时间，需要将原来的
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">3
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;ms
o-font-kerning:
0pt">个人增至
"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;mso-font-kern

ing:0pt">9
宋体;color:black;mso-font-kerning:0pt">个人。

Arial;color:black;mso-font-kerning:0pt">3.
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">重新安排进度。我喜欢
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">P.Fagg
font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerning:
0pt"> , 一个具有丰富经验的硬件工程师的忠告：“避免小的偏差（
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;
mso-font-kerning:0pt">Take no small slips
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">)”。也就是说，在新的进度安排中分配充分的时间，以确保工作能仔细、彻底地完成，从而无需重新确定时间进度表。

Arial;color:black;mso-font-kerning:0pt">4.
10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">削减任务。在现实情况中，一旦开发团队观察到进度的偏差，总是倾向于对任务进减。当项目延期所导致的后续成本非常高时，这常常是唯一可行的方法。项目经理的相应措施是仔细、正式地调整项目，重新安排进度；或者是默默地注视着任务项由于轻率的设计和不完整的测试而被剪除。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;
mso-font-kerning:0pt">

图5

图6

图7

图8

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;

mso-font-kerining:0pt">前两种情况中，坚持把不经调整的任务在四个月内完成将是灾难性的。考虑到重复生成的工作量，以第一种为例（图6 —

论在多短的时间内，聘请到多么能干的两们新员工，他们都需要接受一位有经验的职员

的培训。如果培训需要一个月的时间，那么
宋体;color:black;mso-font-kerining:0pt">三个人月将会投入到原有进度安排以外的

工作中。
宋体;color:black;mso-font-kerining:0pt">另外，原先划分为三个部分的工作，会重新分解成五个部分；某些已经完成的工作必定会丢失，系统测试必须被延长。因此，在第三个月的月末，仍然残留着

mso-bidi-font-family:Arial;color:black;mso-font-kerining:0pt">7

宋体;color:black;mso-font-kerining:0pt">个人月的工作，但此时只有

Arial;color:black;mso-font-kerining:0pt">5

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;

mso-font-kerining:0pt">个有效的人月。如同图

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;

mso-font-kerining:0pt">8

font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:

0pt">所示，产品还是会延期，如同没有增加任何人手（图

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;

mso-font-kerining:0pt">6

font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-kerining:

0pt">)。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;

mso-font-kerining:0pt">期望四个月内完成项目，仅仅考虑培训的时间，不考虑任务的重新划分和额外的系统测试，在第二个月末需要增添

mso-bidi-font-family:Arial;color:black;mso-font-kerining:0pt">4

宋体;color:black;mso-font-kerining:0pt">个，而不是

Arial;color:black;mso-font-kerining:0pt">2

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;

mso-font-kerining:0pt">个人。如果考虑任务划分和系统测试的工作量，则还需

要继续增加人手。到那时所拥有的就不是

mso-bidi-font-family:Arial;color:black;mso-font-ker-ning:0pt">3

宋体;color:black;mso-font-ker-ning:0pt">人的队伍，而是

Arial;color:black;mso-font-ker-ning:0pt">7

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;

mso-font-ker-ning:0pt">人以上的团队；并且小组的组织 and 任务的划分在类型上都不尽相同，这已经不是程度上的差异问题。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;

mso-font-ker-ning:0pt">注意在第三个月的结尾时，情况看上去还是很糟。除去管理的工作不谈，

Arial;color:black;mso-font-ker-ning:0pt">3

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;

mso-font-ker-ning:0pt">月

font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;

mso-font-ker-ning:0pt">1

font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-ker-ning:

0pt">日的里程碑仍未达到。此时，对项目经理而言，仍然存在着很强的诱惑——添加更多人力，结果往往会是上述循环的重复。这简直就是一种疯狂、愚蠢的做法。

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;

mso-font-ker-ning:0pt">前面的讨论仅仅是第一个里程碑估计不当的情况。如果在

Arial;color:black;mso-font-ker-ning:0pt">3

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;

mso-font-ker-ning:0pt">月

font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;

mso-font-ker-ning:0pt">1

font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-ker-ning:

0pt">日，项目经理做出了比较保守的假设，即整个估计过于乐观了，如图

10.5pt;font-family:"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;

mso-font-ker-ning:0pt">7

font-family:"微软雅黑",sans-serif;mso-bidi-font-family:宋体;color:black;mso-font-ker-ning:

0pt">所示。

"微软雅黑",sans-serif;mso-bidi-font-family:Arial;color:black;mso-font-ker-
ning:0pt">6

宋体;color:black;mso-font-ker-ning:0pt">个人手需要添加到原先的任务中。培训、任务的重新分配、系统测试工作量的计算作为练习留给读者。但是毫无疑问，重现“灾难”所开发出的产品，比没有增加人手，而是重新安排开发进度所产生的产品更差。

简单、武断地重复一下Brooks法则：

向进度落后的项目中增加人手，只会使进度更加落后。

Adding manpower to a late software project makes it later.

这就是除去了神话色彩的人月。项目的时间依赖于顺序上的限制，人员的数量依赖于单个子任务的数量。从这两个数值可以推算出进度时间表，该表安排的人员较少，花费的时间较长（唯一的风险是产品可能会过时）。相反，分派较多的人手，计划较短的时间，将无法得到可行的进度表。总之，在众多软件项目中，缺乏合理的时间进度是造成项目滞后的最主要原因，它比其他所有因素加起来的影响还要大。

mso-bidi-font-family:宋体;color:black;mso-font-ker-ning:0pt">