

# 成功项目管理的秘密

摘自 SDMagazine , Karl Wiegers 著 , UMLChina.Shids 译  
模式小组-雷神整理

在最好的情况下，管理软件项目也是很困难的。不幸的是，许多新项目经理实质上没有受到任何就职培训。这里有 20 个成功的管理经验供项目经理参考。

定义项目的标准 .....	2
识别项目的驱动、约束和自由程度 .....	2
定义产品发布标准 .....	2
沟通承诺 .....	2
写一个计划 .....	2
把任务分解成英寸大小的小圆石 .....	2
为通用的大任务开发计划工作表 .....	3
计划中，在质量控制活动后应该有修改工作 .....	3
为过程改进安排时间 .....	3
管理项目的风险 .....	3
根据工作计划而不是日历来作估计 .....	3
不要为人员安排超过他们 80% 的时间 .....	3
将培训时间放到计划中 .....	4
记录你的估算和你是如何达到估算的 .....	4
记录估算并且使用估算工具 .....	4
遵守学习曲线 .....	4
考虑意外缓冲 .....	4
记录实际情况与估算情况 .....	4
只有当任务 100% 完成时，才认为该任务完成 .....	5
公开、公正地跟踪项目状态 .....	5

## 定义项目成功的标准

在项目的开始,要保证风险承担者对于他们如何判断项目是否成功有统一的认识。经常,满足一个预定义的进度安排是唯一明显的成功因素,但是肯定还有其他因素存在,比如:增加市场占有率,获得指定的销售量或销售额,取得特定用户满意程度,淘汰一个高维护需求的遗留系统,取得一个特定的事务处理量并保证正确性。

## 识别项目的驱动、约束和自由程度

每个项目都需要平衡它的功能性,人员,预算,进度和质量目标。我们把以上五个项目方面中的每一个方面,要么定义成一个约束,你必须在这个约束中进行操作,要么定义成与项目成功对应的驱动,或者定义成通向自由的自由程度,你可以在一个规定的范围内调整。相关的详细信息,请参照我的《创建一种软件工程文化》(Creating a Software Engineering Culture)(Dorset House, 1996)中的第二章。

## 定义产品发布标准

在项目早期,要决定用什么标准来确定产品是否准备好发布了。你可以把发布标准基于:还存在有多少个高优先级的缺陷,性能度量,特定功能完全可操作,或其他方面表明项目已经达到了它的目的。不管你选择了什么标准,都应该是可实现的、可测量的、文档化的,并且与你的客户指的“质量”一致。

## 沟通承诺

尽管有承诺不可能事件的压力,从不作一个你知道你不能保证的承诺。和客户和管理人员沟通哪些可以实际取得时,要有好的信誉。你的任何以前项目的数据会帮助你作说服的论据,虽然这对于不讲道理的人来说没有任何真正的防御作用。

## 写一个计划

有些人认为,花时间写计划还不如花时间写代码,但是我不这么认为。困难的部分不是写计划。困难的部分是作这个计划--思考,沟通,权衡,交流,提问并且倾听。你用来分析解决问题需要花费的时间,会减少项目以后会带给你的意外。

## 把任务分解成英寸大小的小圆石

英寸大小的小圆石是缩小了的里程碑。把大任务分解成多个小任务,帮助你更加精确的估计它们,暴露出在其他情况下你可能没有想到的工作活动,并且保证更加精确、细密的状态跟踪。

## 为通用的大任务开发计划工作表

如果你的组经常承担某种特定的通用任务，如实现一个新的对象类，你需要为这些任务开发一个活动检查列表和计划工作表。每个检查列表应该包括这个大任务可能需要的所有步骤。这些检查列表和工作表将帮助小组成员确定和评估与他/她必须处理的大任务的每个实例相关的工作量。

## 计划中，在质量控制活动后应该有修改工作

几乎所有的质量控制活动，如测试和技术评审，都会发现缺陷或其他提高的可能。你的项目进度或工作细分结构，应该把每次质量控制活动后的修改，作为一个单独的任务包括进去。如果你事实上不用作任何的修改，很好，你已经走在了本任务的计划前面。但是不要去指望它。

## 为过程改进安排时间

你的小组成员已经淹没在他们当前的项目中，但是如果你想把你的组提升到一个更高的软件工程能力水平，你就必须投资一些时间在过程改进上。从你的项目进度中留出一些时间，因为软件项目活动应该包括做能够帮助你下一个项目更加成功的过程改进。不要把你项目成员可以利用的时间 100% 的投入到项目任务中，然后惊讶于为什么他们在主动提高方面没有任何进展。

## 管理项目的风险

如果你不去识别和控制风险，那么它们会控制你。在项目计划时花一些时间集体讨论可能的风险因素，评估它们的潜在危害，并且决定你如何减轻或预防它们。要一个软件风险管理的简要的指南，参见我的文章“Know Your Enemy: Software Risk Management”(Oct. 1998)。

## 根据工作计划而不是日历来作估计

人们通常以日历时间作估计，但是我倾向于估计与任务相关联的工作计划（以人时为单位）的数量，然后把工作计划转换为日历时间的估计。这个转换基于每天我有多少有效的小时花费在项目任务上，我可能碰到的任何打断或突发调整请求，会议，和所有其他会让时间消失的地方。

## 不要为人员安排超过他们 80% 的时间

跟踪你的组员每周实际花费在项目指定工作的平均小时数，实在会让人吃惊。与我们被要求做的许多活动相关的任务切换的开销，显著地降低了我们的工作效率。不要只是因为

人在一项特定工作上每周花费 10 小时，就去假设他或她可以马上做 4 个这种任务，如果他或她能够处理完 3 个任务，你就很幸运了。

## 将培训时间放到计划中

确定你的组员每年在培训上花费多少时间，并把它从组员工作在指定项目任务上的可用时间中减去。你可能在平均值中早已经减去了休假时间、生病时间和其他的时间，对于培训时间也要同样的处理。

## 记录你的估算和你是如何达到估算的

当你准备估算你的工作时，把它们记录下来，并且记录你是如何完成每个任务的。理解创建估算所用的假设和方法，能够使它们在必要的时候更容易防护和调整，而且它将帮助你改善你的估算过程。

## 记录估算并且使用估算工具

有很多商业工具可以帮助你估算整个项目。根据它们真实项目经验的巨大数据库，这些工具可以给你一个可能的进度和人员分配安排选择。它们同样能够帮助你避免进入“不可能区域”，即产品大小，小组大小和进度安排组合起来没有已知项目成功的情况。Software Productivity Centre([www.spc.ca](http://www.spc.ca))公司的 Estimate Pro 是可以一试的好工具。

## 遵守学习曲线

如果你在项目中第一次尝试新的过程，工具或技术，你必须认可付出短期内生产力降低的代价。不要期望在新软件工程方法的第一次尝试中就获得惊人的效益，在进度安排中考虑不可避免的学习曲线。

## 考虑意外缓冲

事情不会象你项目计划的一样准确的进行，所以你的预算和进度安排应该在主要阶段后面包括一些意外的缓冲，以适应无法预料的事件。不幸的是，你的管理者或客户可能把这些缓冲作为填料，而不是明智的承认事实确实如此。指明一些以前项目不愉快的意外，来说明你的深谋远虑。

## 记录实际情况与估算情况

如果你不记录花费在每项任务上的实际工作时间，并和你的估算作比较，你将永远不能提高你的估算能力。你的估算将永远是猜测。

## 只有当任务 100%完成时，才认为该任务完成

使用英寸大小的小圆石的一个好处是，你可以区分每个小任务要么完成了，要么没有完成，这比估计一个大任务在某个时候完成了多少百分比要实在的多。不要让人们只入不舍他们任务的完成状态；使用明确的标准来判断一个步骤是否真正的完成了。

## 公开、公正地跟踪项目状态

创建一个良好的风气，让项目成员对准确地报告项目的状态感到安全。努力让项目在准确的、基于数据的事实基础上运行，而不是从因为害怕报告坏消息而产生的令人误解的乐观主义。使用项目状态信息在必要的时候进行纠正操作，并且在条件允许时进行表扬。

这些提示不能保证你的成功，但是它们将帮助你在你的项目上获得一个坚实的把手，并且保证你做了所有你可以做的事来让项目在这个疯狂的世界里成功。

原文链接：[http://www.processimpact.com/articles/proj\\_mgmt\\_tips.html](http://www.processimpact.com/articles/proj_mgmt_tips.html)