

进度管理：项目进度的黑洞 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/283/2021_2022__E8_BF_9B_E5_BA_A6_E7_AE_A1_E7_c41_283684.htm 项目进度控制是项目管理工作中的重要一环，但现在的软件开发项目进度失控的例子却屡见不鲜，甚至进度的延迟总是在快到计划结束的时刻暴露出来，然后谁也不知道到底什么时候才能够结束项目。因此，业内流传着这样一句令人心酸的话：“规划规划全是鬼话，计划计划全是空话”。前不久，我就遇到了这样的一个实际项目。“当进度报告上显示已完成90%时，项目就像遇到了一个黑洞，不断地吞噬着项目组的时间。你说这是怎么了？”在A公司工作的一个好友和我谈起时，话语中露出了深深的不解和抱怨。是呀，问题出在哪呢？根据我的经验，这是经典的“上梁不正下梁歪”问题，我认为要想对项目进度有效的监控与管理，必须抓好以下两个方面：

- q 项目计划：计划的可行性和可操作性是进度监控的基础；
- q 项目进度度量：对项目进度进行科学的度量，才能够获得项目的真实进展情况，并对项目计划做出相应调整。

首先，我们从90%，这个项目完成百分比的来源说起，项目经理在进度报告中写下这个值的时候，他的依据是什么？在这个项目后来的实际情况来看，当时90%的数字是有误的，其实只有50%左右，说明获取这个进度数字时出现了问题。为了更好地理解这个问题，我们来看一个生活中的实际例子：假设我们驱车从厦门开往福州，在途中我们如何获得进度信息呢？对于熟悉这一路段的司机来说这个问题很简单，可以从窗外的景象来得知已经开到哪里，从而做出正确的估计。但是

对于软件开发项目而言，项目团队就像进入了一个全新的征途，就像一个第一次驶过这一路段的司机一样，很难从“窗外的景象”来判断自己的进度。那对于这样的情况，该采用什么方法呢？对于司机而言，他能够通过路边的里程碑这一个简单工具来获知自己的进度信息，那么为什么项目团队不为自己设立一些这样的“里程碑”呢？从这个简单的故事中，我们似乎已经可以得到一些启示，那么现在问题的关键在于如何合理地设立标识项目进度的“里程碑”，接下来我们来看看具体如何操作。在一个软件开发项目中，需要完成的事务很多也很复杂，其复杂度足以让任何人无法对其工作量进行有效的估计，因此对工作任务进行分解是十分重要，这也是设定里程碑的基础。但如何进行工作任务分解呢？这也许也是困扰许多人的一个问题。其实工作任务分解可以从两个方面获得帮助：

q 软件开发生命周期：不管你打算采用什么样的软件开发生命周期模型，它都可以帮助你将整个软件开发项目进行阶段性的划分，而这些阶段就可以做你计划中很重要的里程碑。

q 软件开发需求：软件开发生命周期只给你的项目计划提供了一个框架，而软件开发需求才是其中的血肉，因此软件开发需求的整理与规格化，是细化项目计划的基础。也就是说，在制定项目计划时，应该在你选择的软件开发生命周期模型的框架下，结合软件开发需求来细分任务和设定里程碑。

回顾在这个项目中，他们考虑到项目的复杂性，采用了其熟悉的瀑布型（软件开发生命周期），并且在制定计划时，项目经理认真参考了许多经验值，将2个月的时间按照经验值中的百分比给需求分析、系统设计、编码实现、系统测试、部署交付五个阶段分别安排了时间。并且根

据软件需求说明书的内容，列出了软件模块，并根据每个模块细化了系统设计和编码实现的进度安排。一切看起来都很正常，但是为什么还是没有效果呢？我从他们对细节的回顾中发现了一些问题：q 所有的项目计划均是由项目经理的估计值制定的，也就是说项目经理包办了整个项目计划的制定工作；q 在项目计划中只是简单地在每个阶段的结束时间上标上了一个里程碑符号；q 进度报告中的项目完成百分比，是直接通过“已经历的时间(2个月)”计算得到的；q 项目过程中，需求在变化，但项目计划却没有跟进；q 项目延迟的主要原因在于两个方面：项目需求增加，以及系统设计和编码实现的时间都超过了原先的计划。这一切就是典型的项目进度失控的直接诱因，相信这些项目中都能够发现以上问题的影子。那么如果避免或者解决这些问题呢？在我的资料库中，包括以下几个针对此症的“药方”，在我的实践中收到了良好效果，你也不妨试一试。第一个药方是以面向客户的角度整理需求。我看到许多软件项目开发团队进入了系统设计和编码实现阶段之后，在整个开发团队之间的交流里充满着计算机领域的东西，却难得见到问题领域的东西，这样很容易造成软件开发与客户需求的脱节。因此，从一开始就以面向客户的角度来整理需求，让这些需求的实现成为项目团队共同的目标，这将容易使项目始终保持正确的方向。UML中的Use Case、特征驱动开发中的Feature、极限编程中的User Story都是很好的办法，以这些方式组织的需求，作为项目计划中的血肉，将更有利于进度的安排与控制。第二个药方是项目团队共同完成项目计划。项目计划的一个很重要的前提是项目估算，项目估算最大的基础是经验值，而软件

工程书籍中的经验值反应的只是业界的实践，并不能够反应你的团队，因此在项目估算时应该以自己团队历史经验值为基础，让项目团队中的每一个成员参与估算，这样才能够保证项目计划的可行性。从而避免在本例子中出现的系统设计与编码实现都超出原先的计划的情况。根据我的实践经验，在制定项目计划时，可以将承载着客户需求的用例、特征或用户故事分配给具体的开发人员，让每个开发人员进行估算，并与项目经理进行协商，达成共识。如果每个用例、特征或用户故事太大，以至于估算的时间超过2周，就对其进行细分，直到每个任务块小于2周的时间。最后通过任务卡的形式分配给每一个具体的开发人员，这样就可以保持项目计划与个人的进度计划高度的一致，保证整个项目计划的可行性。

第三个药方是确保里程碑是可验证的。经常看到许多项目计划中，都学会了像模像样地设立了里程碑 在用Project绘制的甘特图中标上一些黑色实心的菱形，但却又发现这些标记这些菱形的项目经理对里程碑的理解是那样的脆弱。最大的问题就在于很多里程碑没有相应的验证标准，也许是有人不小心把公路上的里程碑概念完全带入软件开发项目中来的缘故吧。在软件开发项目中设立的里程碑，其作用是在项目进行时确认进度用的，因此需要给出一个清晰的验证标准，用来验证是否达到里程碑。而验证的标准可以是事件，也可以是工件，例如：“已完成规格化的软件需求说明书的编辑”、“软件需求说明书通过客户签字确认” 可以做为需求分析完成里程碑的验证标准。再如：“已拥有程序的可执行版本，实现了XX 特性，并通过测试” 作为编码实现阶段中的一个子里程碑的验证标准。第四个药方是根据项目的变化动态的更新

项目计划。拿破仑曾经说过，没有一场战争是按照计划打的，但没有一场战争可以在没有计划的情况下赢得的。这句话深刻地诠释了事情发展的动态性，因此在项目开发过程中，项目计划是不可能保持一成不变的。而是应该根据项目的进展，对一些新的需求、新的变化、突发因素做出响应，动态的更新项目计划。例如，当用户提出新的需求时，应该分精力进行分析，做出对项目计划影响的结论，并通过协商与谈判来调整项目计划。而面对动态变化的环境，采用迭代式的使用寿命模型将使你的项目开发团队更好地适应变化。第五个药方是根据里程碑完成情况编写项目进度报告。另外，标识里程碑时，应该同时赋予里程碑标识进度值的功能，更通俗地说，就是让每个里程碑带上一个百分比，告诉团队通过这个里程碑说明项目完成了多少。这样，项目进度报告上的完成百分比将显得更加真实和有意义。当然随着项目计划的动态变化，未到达的里程碑的这一值也应该做出相应的调整。

项目管理是一个实践学科，理论指导只能够帮助你正确地做事，为你的实践提供一些方法与指南，却无法保证项目成功。“银弹”并不存在。在实践中去发现问题、解决问题、总结思路，才是最有效的办法。最后，也奉劝每一个项目管理者，不要迷信标识进度的“已完成百分比”，从黑洞中逃离。

项目背景：A公司前不久接下了一个客户化开发的软件项目，其主要的内容是为一个汽车销售公司做一个MIS系统，并且要求整个项目将在2个日历月之内完成。合同签署之后，该公司指派了一名项目经理，该项目经理看上去也十分认真，在经过了需求调查之后，他就向公司提交了一份详细的项目计划书，而且项目完成的时间也完全与合同要求相同

，整整2个日历月，一切看起来是那樣的順利。時間過得很快，項目似乎也進展得很順利，項目經理也嚴格按照規定每周上交了用Project 2000制作的漂亮的進度報告，項目完成的百分比也一直和項目計劃保持着一致，很快到了第八周，項目進度指示已完成90%。但是，第九周出了問題，項目無法按時交付，希望能夠再延長兩周。A公司的市場部門急了，你不是上周就完成了90%嗎？這周出了什麼問題！項目經理解釋說，項目的需求一直有變化，增加了不少工作量。沒辦法，市場部門開始向客戶解釋。兩周過去后，進度報告上指示完成了94%，希望能夠再延長兩周。這時候不僅是市場部門火了，客戶也氣急敗壞。但是，這並沒有解決問題，項目一直拖到了4個日历月才完成，延期交付給A公司帶來很大的經濟與信譽損失。100Test 下載頻道開通，各類考試題目直接下載。詳細請訪問 www.100test.com