

项目管理实施体会（二）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/64/2021_2022__E9_A1_B9_E7_9B_AE_E7_AE_A1_E7_c41_64823.htm IV. 产品的需求、系统分析、设计总的而言，软件开发主要分为六个阶段：需求分析阶段、概要设计阶段、详细设计阶段、编码阶段、测试阶段、安装及维护阶段。开发软件就如八股文一样：总体规划、项目立项、需求分析、系统分析、系统设计、编码实现、项目测试、文档制作（八股文：破题、承题、起讲、入手、起股、中股、后股、束股），一切都按部就班。同理，信息系统集成项目管理一般的过程分为：需求分析、项目调研、方案论证、实施方案、具体实施、测试、验收、售后服务。同时项目管理按管理的方式可以分为售前、售中和售后。在国内，做的项目越多，就越容易产生这样的感觉：项目感觉总是做不完，就像一个“无底洞”。用户总是有新的需求要项目开发方来做，就像用户在“漫天要价”，而开发方在“就地还钱”。实际上，这里涉及到一个“范围管理”的概念。项目中哪些该做，哪些不该做，做到什么程度，都是由“范围管理”来决定的。“范围管理”的相关知识，可以参考项目管理知识体系指南、项目管理九大知识体系的相关内容。这里只说说两点，核心需求和需求的变更。一般说来，需求对用户来说，可以分为核心需求和辅助需求。对于中国贪心的用户来说，功能从来不嫌多的。核心需求就是用户使用本软件是，一定会使用的功能，没有这些功能，会影响用户的忠诚度。辅助功能，就是用户可用可不用的功能。有了这些功能，用户更喜欢，没有这些功能，用户不会太在意，或

者是可以容忍。同样一个软件，不同的用户，核心需求和辅助需求是不一样的。在一个项目产品中我们应该知道对方需要什么，自己要做什么，这是项目成功的基础所在。对于产品，因为用户是不确定的，确定哪些需求是核心需求，哪些是辅助需求，则比较难。但是如果定位合理，对定位的用户范围的特征分析得准确，则不是什么太难得事。但无论是产品还是项目，都必须考虑以下三点：1、对于用户提出的每个需求都要知道“为什么”，并判断用户提出的需求是否有充足的理由，并考虑由此而引申出来的问题，触类旁通，举一反三；2、对于用户提出的每个需求都要知道“为什么”，并判断用户提出的需求是否有充足的理由；3、分析由用户需求衍生出的隐含需求，并识别用户没有明确提出来的隐含需求（有可能是实现用户需求的前提条件），这一点往往容易忽略掉，经常因为对隐含需求考虑得不够充分而引起需求变更。需求确立后，重要的是规范化，文档化，可参考常用的需求建模的方法如数据流图（DFD）、实体关系图（ERD）和用例图（Use Case）三种方式。需求的变更是不可避免的，如何以可控的方式管理软件的需求，对于项目的顺利进行有着重要的意义。现在，重要的不是限制需求的变更，而是让用户、销售、开发都明白，不管是哪种情况下变更，只要是造成变更的一方，都要背负相应的责任。这就需要一个明确的约束机制。在制定制度时，大家都可以参议，提意见。但只要制度确定了，就必须恪守这个制度，是制度管人，而不是领导管人。也只有这样，才不至于开发处于遥遥无期的状态，而到最后则草草收兵。本人认为，需求的确定可以分为三个阶段则比较合适，这样划分也涉及到计划、成本的管理

，可参考后面的相关内容。一、项目初期，用户、销售、开发三者，应根据WBS方法确定整个项目的整体需求，即最起码要确定WBS中最顶层的父作业，如需求一，需求二．．．．．并尽可能详细地确定需求地具体划分，如需求一.1，需求一.2，需求二.1，需求二.1.(1)．．．．．对于开发周期比较长的项目，则可以确定每个需求开发开始的大概时间，周期比较短的项目，则只需确定整个项目的开始时间即可。对没有确定的需求，则要明确在开发前一个相应的时间，必须在确定的时间之前来落实详细的需求。对已确定的需求则明确在开发前的一个具体时间内可以修改的内容，主要是框架性的需求。二、项目开发前，核实项目的所有需求，应当包括详细的需求，并明确告知用户以后只能修改细节上需求，主要是操作上的需求。如果在以后要修改框架性的需求，必须受到一定的约束，所产生的成本必须为要变更的一方负责。如是因为开发的原因造成的，则增加成本纳入到项目的成本中；如因销售工程师因销售压力而盲目对用户作承诺的功能，则相应成本必须纳入销售成本中；确定落实需求后，进行项目的开发阶段。三、项目开发完成后，提交给用户试用。用户使用后，用户明确要修改的内容，主要范围是操作上，细节的实现是否与需求一致，是否与用户真正需要的一致。如要对框架性的需要修改，则必须按先前的约束条件进行。确定修改内容后，向用户明确，以后的修改内容都只能局限于当前提出的修改范围，当然，程序出错除外。开发进行修改，用户使用，将修改范围进一步缩小。一次或多次迭代后，项目完成。V.工作量的估算及评价 项目管理最大的难度，就是每一模块的工作量、开发时间的确定，这也是

项目实施的主要风险，最难预测、控制的风险。比较常用的估算方法有：估计项目交付日期的方法有很多，如基于经验的估计、基于模型的估计等。一种简便易行的估计方法是采用Wideband Delphi估计方法，此方法可以降低不同人员所作估计的偏差。基于模型的估计方法则包括KLOC、FPA以及COCOMO 等模型。在这里主要引用新的估算方法，以供参考。如果，公司有着类似项目实施的丰富经验，则工作量、开发时间的确定则会更切实际。首先，提取公司内部数据，统计出类似模块的工作量（M公）、开发时间（S公）。如果，公司没有这方面的经验，那如何办呢？不做这一步？！对，没办法，没时间时也可以放弃这一步。有时间，有精力时，可以通过了解社会对类似模块的工作量、开发时间的平均值再结合本公司的情况进行假设模块的工作量、开发时间。其次，项目开发成员对这一模块的工作量、时间的估算。这一步是最耗时，这是很多公司都没有去做的，包括国内一些知名的软件公司。软件开发，不是工厂中的流水线生产，可以对产品的生产过程中每一个过程，每一个步骤，进行严格的控制和限制。在开发过程中，由于开发人员不同的学历背景、知识水平、经历、开发经验、对模块的理解程度、思维方式、编程习惯等因素，对同样的模块，所需求的工作量，开发时间是有很大大区别的。因此，需要每个开发成员对相关模块了解熟悉后，进行估计，再根据不同的系数，通过不同的公式进行转换后确定每一个模块的工作量、开发时间。

公式为： $M = M(公) * X1 M(项) * X2$ 。 $S = S(公) * X1 S(项) * X2$ 。 $M(项) = M(开1) * Y1 M(开2) * Y2 \dots M(开n) * Y.n$ 。 $S(项) = S(开1) * Y1 S(开2) * Y2 \dots S(开n) * Y.n$ 。每一模块，每项功

能完成后，对小于原定工作量或成本的，可以直接以原工作量和成本作为考核的依据。对于超出原定工作量或成本的，必须组织相关人员进行成本、工作量的评估。主要的操作方法为，组织3到5人的评估小组，小组成员尽可能是在开发时对要评估模块相当熟悉的成员。小组成员利用一定的时间了解评估模块的代码，功能特点，模块实现思路，难点，简单的单元测试等，然后小组的成员假设是成员本人来开发评估模块，需要用的成本、工作量并转换为开发人员相应级别的工作量、成本，最后综合小组成员的估计的工作量、成本和开发人员开发时所耗费的成本、工作量进行加权平均，得出最后的成本、工作量，并以此作为考核的依据。100Test 下载频道开通，各类考试题目直接下载。详细请访问

www.100test.com