项目管理的项目可控性讨论

040630520 彭立勋

项目管理中两个重要的内容是进度可控和质量可控。项目进度可控，可以依靠项目生命周期中有三个与时间相关的重要概念，这三个概念分别是： 检查点（Checkpoint）、里程碑（Mile Stone）和基线（Base Line），它们一起描述了在什么时候（When）对项目进行什么样控制。项目质量可控，则是靠建立完善的流程规范，从分析到建模，从编码到测试，再到最后的交付，都要有一套质量可控的规范在运作，它们描述什么人（Who）需要再项目的每个部分做什么（What）。

有了 3 个 W（When/Who/What），就明确了什么人在什么时间做什么事，就能够保证项目的可控。虽然项目的成功依赖的因素远比这些多，但有了可控，至少能保证在规定的时间内现有技术可以解决的问题都可以被实现。

➢ 进度可控

项目进度可控，可以依靠项目生命周期中有三个与时间相关的重要概念，这三个概念分别是： 检查点（Checkpoint）、里程碑（Mile Stone）和基线（Base Line），它们一起描述了在什么时候（When）对项目进行什么样控制。

1. 检查点

检查点指在规定的时间间隔内对项目进行检查，比较实际与计划之间的差异，并根据差异进行调整。可将检查点看作是一个固定“采样”时间点，而时间间隔根据项目周期长短不同而不同，频度过小会失去意义，频度过大会增加管理成本。常见的间隔是每周一次，项目经理需要召开例会并上交周报。

例如在我实习的公司，每天都是一个检查点，需要上报每天的工作日志，总监可以根据工作日志来判断每个人的工作情况，对工作较慢的成员进行催促，保证整体进度。然后每周提交任务完成情况，总监根据完成情况判定成员是否努力工作了。将大任务分解成小任务，小任务再分解成可查看的小目标，这样一步步控制，在最小的地方及早发现进度问题，及时催促，保证了项目整体进度不拖延。

2. 里程碑

完成阶段性工作的标志，不同类型的项目里程碑不同。里程碑在项目管理中具有重要意义，用一个例子说明：

情况一：让一个程序员一周内编写一个模块，前 3 天大家可能都挺悠闲，可后 2 天就得拼命加班编程序了，而到周末时又发现系统有错误和遗漏，必须修改和返工，于是周末又得加班了。

情况二：实际上还有另一种选择，即周一与程序员一起列出所有需求，并请业务人员评审，这时就可能发现遗漏并即时修改；周二要求程序员完成模块设计并由项目经理确认，如果没有大问题，周三、周四就可让程序员编程。同时项目经理准备测试案例，周五完成测试；一般经过需求、设计确认，如果程序员合格则不会有太大问题，周末可以休息了。

第二种方式增加了“需求”和“设计” 两个里程碑，这看似增加了额外工作，但其实有很大意义：首先，对一些复杂的项目，需要逐步逼近目标，里程碑产出的中间“交付物” 是每一步逼近的结果，也是控制的对象。如果没有里程碑，中间想知道“他们做的怎么样了” 是很困难的。其次，可以降低项目风险。通过早期评审可以提前发现需求和设计中的问题，降低后期修改和返工的可能性。另外，还可根据每个阶段产出结果分期确认收入，避免血本无归。第三，一般人在工作时都有“前松后紧”的习惯，而里程碑强制规定在某段时间做什么，从而合理分配工作，细化管理“粒度”。

3. 基线

指一个（或一组）配置项在项目生命周期的不同时间点上通过正式评审而进入正式受控的一种状态。基线其实是一些重要的里程碑，但相关交付物要通过正式评审并作为后续工作的基准和出发点。基线一旦建立后变化需要受

控制。

重要的检查点是里程碑，重要的需要客户确认的里程碑，就是基线。

在我们实际的项目中，周例会是检查点的表现形式，高层的阶段汇报会是基线的表现形式。

➢ 质量可控

项目质量可控，则是靠建立完善的流程规范，从分析到建模，从编码到测试，再到最后的交付，都要有一套质量可控的规范在运作，它们描述什么人（Who）需要再项目的每个部分做什么（What）。

1. 需求分析

需求分析指的是在建立一个新的或改变一个现存的电脑系统时描写新系统的目的、范围、定义和功能时所要做的所有的工作。需求分析是软件工程中的一个关键过程。在这个过程中，系统分析员和软件工程师确定顾客的需要。只有在确定了这些需要后他们才能够分析和寻求新系统的解决方法。

只有通过正确的需求分析，才能提炼出客户需求中的对象及数据，以及功能，这是进行开发的关键所在，保证需求分析的质量是保证整个软件工程质量的基础保证。

2. 系统建模

软件工程中的系统建模是指，将需求分析所得到的系统中的对象、数据、功能进行合理的设计、整合，得出满足客户需求的以用代码实现的系统模型。

系统建模的质量决定了系统实现出来的质量，可扩展性、高可用性、系统整体效率都是在系统建模这一层就已经固化，因而系统建模是保证软件质量的最核心部分！

3. 程序编码

无论再好的设计，都是需要靠最终的代码来实现，编码的质量和规范，决定了代码的可读性和可修改性已经程序的效率。

编码中需要有注释规范、排版规范、命名规范、优化规范等等组成，有了这些规范，才能保证每个参与系统开发的人之间代码可以交流，有良好的可维护性。

例如在我实习公司的开发中，我们指定了通用于 HTML、JavaScript、Java、PHP、SQL 等语言的标准命名规范和标准注释规范，这样即使跨不同的语言可以读懂变量及函数的含义，具有两好的可读性，保证代码可维护。另外我们还制定了适用于各种不同语言的排版规范，通过良好的排版，代码的可读性会大大增加，后来的维护人员更容易看懂代码。

4. 系统测试

系统测试是项目进程中的重要部分，测试是保证代码质量的最后一关，编码过程中出现的问题都靠测试环节进行检查。

系统测试分为：单元测试、功能测试、系统集成测试。

单元测试是在软件开发过程中要进行的最低级别的测试活动，在单元测试活动中，软件的独立单元将在与程序的其他部分相隔离的情况下进行测试。

功能测试是根据产品特征、操作描述和用户方案，测试一个产品的特性和可操作行为以确定它们满足设计需求。本地化软件的功能测试，用于验证应用程序或网站对目标用户能正确工作。使用适当的平台、浏览器和测试脚本，以保证目标用户的体验将足够好，就像应用程序是专门为该市场开发的一样。 功能测试也叫黑盒子测试或数据驱动测试，只需考虑各个功能，不需要考虑整个软件的内部结构及代码。一般从软件产品的界面、架构出发，按照需求编写出来的测试用例，输入数据在预期结果和实际结果之间进行评测,进而提出更加使产品达到用户使用的要求。

集成测试是在单元测试的基础上，将所有模块按照设计要求，如根据结构图，组装成为子系统或系统，进行集成测试。实践表明，一些模块虽然能够单独地工作，但并不能保证连接起来也能正常的工作。程序在某些局部反映

不出来的问题，在全局上很可能暴露出来，影响功能的实现。

在我实习公司的开发中，主要进行单元测试来保证代码质量，功能测试由业务人员参与，集成测试难度太大，没有进行。

5. 系统交付

在所有的测试完成之后，就是最终的交付了。系统的交付也是需要规范的流程，打包的方式，部署的形式，版本控制，后期维护等等都需要全面的考虑，才能保证客户获得的是高质量的软件。

我在我实习的公司进行的团队规范改革：