论项目的范围管理

[摘要]

  2015年5月开始，我参加了公司面向17年XXXXXXX的嵌入式项目研发。该项目主要是在15年XXXX的基础上，增加XXX功能，XXX功能，XXX功能等新商务机能，目标在17-18年进一步扩大欧美高端SOHO市场占有率。该项目光嵌入式软件开发规模大约在500人月左右，由总公司和中国研发中心（虚拟团队）共同开发，因本人有多年XXX嵌入式行业开发经验，被任命担当中国方项目经理一职。**众所周知大项目的项目范围管理是一项至关重要的管理工作,往往决定着项目的成败。**本文主要阐述的是项目的背景，项目范围管理过程中遇到的问题和困难，在导入XDDP派生开发流程的背景下，**分别说明了收集需求，定义范围，做成WBS，确认和控制范围等各个范围管理子过程**中，我是如何克服困难，并使项目如期验收通过取得成功。在本文最后，也总结了自己在项目范围管理上的一些不足。

[正文]

项目概述：

  为了在17-18年度进一步扩大欧美高端SOHO市场的占有率，总公司决定在15年的开发基线上，增加新的功能。该项目光是软件部分的开发规模大约在500人月左右，其中近三成的工作量决定实施中国外包。该项目的开发工期是15个月，采用C语言在原来产品的基础上进行派生功能开发，以完成新商务机能。因该项目的特殊性，在被任命为项目经理以后，我决定导入XDDP软件派生开发流程来做这个项目。XDDP由日本人清水吉男桑创建的软件开发流程。它适用于面向现有系统进行设计变更以及增加新功能时而设计的派生开发流程。

**项目范围管理是项目管理中至关重要的管理知识领域，项目范围管理主要包括：收集需求，定义范围，制定WBS，范围确定，范围控制等过程。**因本次嵌入式软件委托开发项目的研发规模大，项目建设期长，需要总公司和中国受委托开发团队之间的协同开发，作为中国受托开发的虚拟团队，于是就有了下面**三个在范围管理上的比较有代表性的困难。其一，在无法直接与客户进行F2F沟通，需求暧昧。其二，各个设计需求之间范围影响不同，难免会产生前后矛盾。其三，该项目成果物繁多，为范围确认与逐一验收带来了阻碍。**

针对上面三个困难，在本项目的范围管理各个过程中，我主要做了以下的工作。

1． **收集需求和范围定义**

  因本次新追加了功能都已经在XXXX上实现，其他小功能也均在从其他不同的产品线上实现功能，所以需求就是一句话，让我们参照既有产品功能，然后将功能全部移植到XXXX上。但XX和XXXX机属不同的代码基线库，且原理的不同，所以是无法做到100%功能移植的。这就给我方收集需求上带来了困难。

**针对上面所说的需求分析的困难，我采取了分析产品，系统交互图，引导式研讨会等技术获取和确认需求。**我从公司项目资产库中调取了XXXX机参照功能的机能说明书，结合测试报告书。分析完现有的文件后，调来了实机，和团队成员一起根据测试报告书，一起在实机上动作了一下需要移植的功能，在充分理解机能以后，使用UML测试用例图对功能进行了建模，同时也要求对每张图都要明确系统输入和输出。在团队内部评审后，我便根据干系人登记册的内容，逐个功能与干洗人召开需求讨论会，在用例图的基础上讨论定义的机能需求。同时重点说明了经过我方可行性分析后，因框架的差异无法移植实现的需求。在视频会议上引导干系人听取我方的意见后，得到了干系人关于需求的详细反馈。在稍作修改后，各个功能的需求逐渐清晰，在UML用例图的基础上，做成了需求文件列表，然后进行了范围定义，做成了范围定义说明书。在发给干系人得到了承认后，收集需求和范围定义的工作圆满结束。

2． **创建WBS(工作分解结构)和变更设计TM**

**创建工作分解结构是把项目可交付成果物和项目工作分解成较小的，更加易于管理的组件的过程。**因本项目建设规模大，树形结构的WBS不能很好的使用，所以需求的分解使用了列表形WBS。根据项目的特殊性和每个需求的特点，我和团队成员一起在分解需求的同时，加上了每个需求的背景和理由并把这些内容一并写入并作为了项目的范围基准。正式进入开发阶段后发现了一个问题，各个设计需求之间范围影响不同，有些需求在考虑设计实现的过程中，会产生前后矛盾的情况。一般情况下如果确认了一个变更点之后马上修改代码，这样就很可能会没有注意到模块之间的相互影响和对产品现有功能的影响，而造成返工。针对这个问题，我引入了XDDP派生开发中的变更设计TM来解决。

  变更设计TM是XDDP派生开发的重要可交付成果物，它的格式类似于需求跟踪矩阵，每一行登陆分解后的树状结构WBS，与根据矩阵不同的是每一列登陆所需要修改的源代码文件的名字。它要求针对每一个变更内容，不要马上进行修改，首先将要修改的思路根据将要修改的源文件的不同写入到变更设计TM里面，通过自我评审和团队内部评审来充分分析变更设计所产生的对现有代码的影响范围，在变更设计TM的做成过程中，出现了多个需求改同一个C源文件甚至是同一个函数的情况，通过作成变更设计TM，在未进入编码阶段前，就清楚地看到了各需求之间相互影响，与现有功能的冲突部分，因为用分解后的WBS定义了变更设计TM，且用该文件也较好地跟踪了需求的实现情况，各个设计需求之间范围影响而导致质量下降的问题并未出现在本项目中。

**3.范围确认和控制**

  因为项目周期长，规模大待实现的功能和需要交付的成果物繁多，变更需求小而多，如果不能妥善安排项目的验收确认过程，将会在项目收尾时出现混乱情况。针对这个问题，我使用变更设计TM来完成需求实施情况跟踪的同时，采用三段式验收方法，即阿尔法版，贝塔版和最终版来实施成果物的验收。因为每个需求变更对应一张JIRA票，于是每完成一个需求，在将成果物记到JIRA后，要求客户方负责人在一周内确认验收。验收有问题通过JIRA联络。验收完再JIRA上打上标签以便日后查找。使在各个子需求子功能顺利验收后，更新变更设计TM。然后在关键里程碑上再次核对成果物采取联合验收，为最终项目的验收提供有效保障。

  在范围的控制上，我使用JIRA票来管理变更。将实施变更管理流程的过程和结果都分批次反映到JIRA票上。我将变更分成了三类，新增功能，既有功能变更，完善性变更(代码重构)，并在JIRA上定义优先级，根据优先级高低，决定阿尔法版验收时间，同时在JIRA票上定义好子需求验收的方式。有些完善性的内部变更，因为考虑到项目整体情况范围蔓延和质量镀金等因素，和客户方协调后更改为了次机种检讨项目。

[结束语]

  由于项目范围管理得当，和团队成员一起努力14个月以后，本项目于16年8月顺利完成最终合同验收并取得客户的一致好评。但是也有一些问题，例如在收集需求过程中，团队中缺乏有既有系统分析能力的组员，且没有提前识别并安排培训，导致收集需求和定义范围在启动阶段花了很多工时。在发生变更时，因为分JIRA票分开来管理变更需求，所以显得有些杂乱，也出现了未及时更新范围基准的情况。**这些范围管理的问题我会在以后的项目中加以借鉴和改进。**