

# 电子 期刊

作者：思步网会员

主编：jane

校对：steplv

**第十一期**

封面设计：steplv

发布：2008.10.18

## 著作权说明

本文档为免费电子文档，任何人可以在思步网<http://www.step365.com> 免费下载。**著作权属于思步网及思步网会员共同所有。**

在不对本文档做任何修改的前提下，任何人都可以在互联网上自由下载、传播本文档，也可以放在自己的站点供他人下载。

若通过互联网在线转载其中部分内容，或者通过其他媒体转载本电子书及其中部分内容，必须注明文档来源“思步网 <http://www.step365.com>”和文档作者。

本文档仅供学习交流之用，未征得思步网及思步网会员的同意，本文档不得用于商业用途。

由于编者水平所限，在文档中难免会出现错误，欢迎读者对本文档提出批评建议，意见请反馈至：[service@step365.com](mailto:service@step365.com)，谢谢！

## 阅读注意事项

在阅读本电子文档时，请留意以下字符（字母）所代表的含义。

本电子文档第一部分中：

**Q** (user name)：Q 代表问题 Question

**A** (user name)：A 代表解答 Answer

**C** (user name)：C 代表结论 Conclusion

括号中内容 user name 为思步网会员名

例如：**Q** (steplv) xxxxx, **A** (sungubbi) zzzzz, **C** (fishred) yyyyyy

表示：

提出问题者为会员 steplv，问题内容为：xxxxx

解答问题者为会员 sungubbi，解答内容为：zzzzz

做总结者为会员 fishred，总结内容为：yyyyy

依次类推。

## 思步在线交流

✧ WebSite: [www.step365.com](http://www.step365.com)

为了方便广大网友、会员及时地交流互动，思步网除了提供在线的留言交流（含论坛、日志、圈子、资源、招聘等）外，还专门建立了实时的 IM 交流群，目的是让大家更方便、快捷地获得第一手的探讨交流信息，实现“面对面”的交流效果。

为了突出不同实战经验者分享、交流心得体会以及探讨相关的疑问，我们将群分为三大类：

### 1、Step365.COM-江湖群

少于 3 年工作经验者，可加入**江湖群**（意即**初混江湖-学徒者**）。主要定位：**每日问题答疑，网友有咨询师共同解答；提供入门思步刊物。**

- MSN群：

Step365.COM-江湖群①: [mgroup24287@hotmail.com](mailto:mgroup24287@hotmail.com)

- QQ群：

Step365.COM-江湖群②: 19894513（未满员，申请加入时，请注明职业）

Step365.COM-江湖群①: 43160269（已经满员，不能再加入）

### 2、Step365.COM-侠客群

大于 3 年工作经验者，可加入**侠客群**（意即**除危济困-英雄者**）。主要定位：**每周一个技术解决方案讨论；专业实践丰富咨询师，每日随时解答疑问。**

- MSN群：

Step365.COM-侠客群①: [group103008@msnzone.cn](mailto:group103008@msnzone.cn)

- QQ 群：

Step365.COM-侠客群①: 56397267

### 3、Step365.COM-小憩群

为满足大家灌水的需求，避免在“江湖群与侠客群”乱聊一些不相干的话题，特别开通“**小憩群**”。定位：天上、地下、天文、地理、人文、男男女女、吃吃喝喝、潇潇洒洒、快意人生……等等，只要能想的到的，都可以乱聊。

- MSN群: [group134479@xiaoi.com](mailto:group134479@xiaoi.com)

## 目 录

著作权说明.....	2
阅读注意事项.....	3
思步在线交流.....	4
第一部分：思步简讯.....	6
第二部分：你问我答.....	7
1 全员改进过程.....	7
★ 习惯的力量.....	7
2 全员改进案例.....	8
★ 缺陷相关，请达人释疑.....	8
3 项目管理.....	11
★ 项目管理从猴子说起.....	11
★ 项目经理=什么.....	12
★ 如何进行质量分析.....	15
4 QA/EPG专区.....	16
★ 项目中代码评审没执行——谁的责任？.....	16
★ SQA如何收集度量数据，建立有效度量体系？.....	17
5 CM专区.....	19
★ 配置管理带来的好处.....	19
★ 备份记录的问题探讨.....	20
第三部分：会员原创.....	21
1 CMMI方面.....	21
★ 实施CMMI时必须解决的认识问题.....	21
★ 研发过程如何进行裁剪.....	23
2 TEST方面.....	26
★ 测试方案和测试计划的区别.....	26
3 RD&RM方面.....	28
★ 需求编写的几点经验之谈.....	28

## 第一部分：思步简讯

### 思步快讯：开通“思步译站”专栏

本着“传播国外先进理念、提升会员英语读写能力”的宗旨，特别开通“思步译站”专栏。

该专栏主要是“国外英文资料的中文化”，通过会员们之间的协同努力，在学习国外理念的同时，又提升英语阅读与翻译的水平，双重收获。

目前，该专栏将进一步增加人员力量，期待更多的同仁加入我们的队伍。让我们在知识中成长；让我们在成长中学习；让我们在学习中进步；让我们在进步中收获人生。

详情请登录思步网（<http://www.step365.com>）、及思步协作服务论坛，查看相应的公告、专栏与说明。



## 第二部分：你问我答

### 1 全员改进过程

#### ★习惯的力量

##### Q (steplv):

说说“习惯”在过程改进中的“作用”。

大家有没有同感，我们总是习惯于说：我以为。

“我以为”这三个字，在总体上，已经代表了习惯的意思，我们总是在固有的思维定势来判断事物的好坏与难易，却很少停下来去思考这背后所应该蕴藏着的含义。

我本人也是这样，在今天以前，我往往会说：我以为。但从今天以后，我想我应该摒弃这种“我以为”的毛病，我要检讨。

话说回来，回到这个主题的本身：习惯的力量！没错，人们都不应该去忽略习惯的力量，这种力量往往是可怕、可畏的。

正如我们在公司做改进一样，为什么一个新生的事物（流程或者规则）总是这么难以去执行与推广，我想，在这期间充满了太多的“习惯的力量”的抵触。在这种抵触情绪中，不乏有公司的高层领导，所以，改进工作是推一回、难一回，最后就不了了之了。究其原因，我想除了企业本身文化的“熏陶”外，更多的则是人们习惯难以打破的原因。

在这个时候，“习惯”就变成了一种阻碍，习惯力量的强弱，则决定了改进工作的难易。毋庸置疑，改变一种习惯是何等的不易。

如果打破习惯，针对于过程改进工作来说，就显得太重要了。那么，大家可以继续展开探讨，如何破除过程改进路上这种“坏习惯”的障碍，顺利推进过程改进工作。

同时，我又想到了之前发的一个帖子：唐僧师徒四人行。期待更多的人加入探讨。

##### A(jaedonger):

其实，有时候习惯有时候没什么不好。

CMMI 不正是总结前人的，然后总结的嘛。

“唐僧师徒四人行”中，总不能指望唐僧去斩妖除魔吧（他的功能是最大的发挥手下人的作用）；也不用指望“齐天大圣”去赶沙僧的挑担，那发挥不出他的本事，属于大材小用了。

以上愚见，不要拍砖。

A(step365he):

习惯是什么?

一是指常常接触某种新的情况而逐渐适应

二是指在长时期里逐渐养成的、一时不容易改变的行为、倾向或社会风尚

三是指生活中相对稳定的部分

四是指以相同的方式，一而再，再而三地从事相同的事情——不断重复——不断思考同样的事情。

而且，当习惯一旦养成之后，它就像硬化了的水泥块，很难打破了。

可见习惯是个中性词，看你怎么运用了。

C(jane):

习惯的作用，并不能三言两语能讲清楚。不过用在过程改进的确很合适，过程改进其实就是一个养成习惯的过程，其实好多时候在推新的流程时，遇到这样那样的问题。基本可以用习惯来解释，当大家慢慢接受，也就是一个习惯在慢慢形成的过程。当然习惯也分好习惯和坏习惯，所以在定义习惯时的要求远远比养成习惯要难得多。如果制定了好的习惯，就别怕大家的抵触，因为迟早会收到效益!!!

## 2 全员改进案例

### ★ 缺陷相关，请达人释疑

Q(justese):

如果在软件开发过程中引入了较多的缺陷那么发行的系统中也存在较多的缺陷。

在各种资料中经常看见类似的话，有几点疑问：

- 1.这句话是对所有的项目都可以这样说还是对一部分项目正确?
- 2.这个结论是怎么得出来的?如果是通过度量数据，那么这些数据都是哪些数据?

A(sungubbi):

开发过程中引入了较多的缺陷，有两种情况，一是开发确实很糟引入了BUG，二是测试很强、尽可能多的发现了问题。

理论上来说，修复缺陷将可能引入更多的缺陷，但是在上线前发现越多的问题，确实可以减少发布后用户发现的缺陷数量。但是，每发现一个缺陷，在开发、测试、构建等各



环节需要投入的工作量也会更大，所以最重要的是将缺陷消灭在编码工作中。

所以，你的两个疑问，我认为：

1.这句话是对所有的项目的都可以这样说还是对一部分项目正确？

不完全正确。如果相比较两个产品来说，通常情况下，开发过程中 BUG 越多的那个产品，则上线后的 BUG 一般也较多。对于同一个产品，如果测试发现的缺陷被修复，则发布后的缺陷通常会减少。如果是测试不充分而未发现多少问题，并不代表上线后的缺陷数量就会减少。

2.这个结论是怎么得出来的？如果是通过度量数据，那么这些数据都是哪些数据？

上线前缺陷与上线后缺陷存在一些联系。你可以通过上线前的缺陷密度与上线后的缺陷密度来判断某一个产品的缺陷情况。也可以通过这个数据来比较两个产品的缺陷密度，是不是上线前缺陷密度越高的那个，在上线后缺陷密度也是偏高的。

A(step365he):

1. 牵扯到另一个话题——质量成本。

如果不重视过程中的各项预防缺陷工作，那测试成本会大大增加，而且发行后导致的用户不满意度导致的退货或者升级付出的额外开发成本也是无法估量的。

如果重视过程中的预防缺陷工作，暂时性质量成本会增加，随着预防越陷卓有成效，导致测试成本降低，也会使得之后额外成本在可控范围内。

2. 如果一个产品开发中问题多多，即使测试人员很强，在初期一段时间，辛勤工作后将缺陷尽可能的消灭，保证了产品发行质量。随着这类情况周而复始，他们会陷入一种无力感，最后测试人员会流失。。

A(justese):

还有一个疑问：

“发行后系统中的缺陷”这句话中的缺陷是指发行后一段时间内，用户在使用中发现并反馈回来的缺陷，还是指所有的缺陷（包括用户发现的缺陷以及系统中隐藏未被发现的缺陷）。

如果只是指用户使用中发现的缺陷，我比较清楚，我带过几个项目，也参加过测试工作，根据度量数据可以得出结论，开发过程中 BUG 越多的产品，则上线后的 BUG 也较多。到最后用户不满意以至退货。

但是如果是指所有的缺陷，这种说法我不认同。既然是隐藏的，那么我们就不知道它究竟有没有，有多少，没办法定量的分析。那也就不能得出结论。

头好乱啊，想得越多结就越多，理不顺了。

A(iamredeye):

1. "quality is designed in, not tested in." 测试很大程度上只是一种测量手段，试图测量出软件到底有多少 defect。但由于测试没有穷尽，也许我们永远都无法知道产品到底有多少 defect，但是上帝知道这个客观上存在的数字！测试结果从统计角度讲已经“证明”了产品的质量—并且一定程度上进行了量化—defect number, severity, etc

2. 所以开发阶段的“测量”结果同样可以用来推断上线之后的质量。即使随时间质量有一些改进，但基本不会有质的变化。

A(step365he):

“指所有的缺陷（包括用户发现的缺陷以及系统中隐藏未被发现的缺陷）”——这句话有悖论：隐藏的并且是未被发现的。既然你都没发现，怎么知道是隐藏的？你所能确定的只有“测试无穷尽”。因此在度量的范围只能限于已被发现的（无论是哪个干系人发现）。

A(iamredeye):

测试无穷尽，但却有迹可循。这就是为什么好的 test designer 比较少。通过有限的样本测试推测出全貌，未必精确，但可以足够准确。

A(step365he):

1. 软件测试中存在杀虫剂现象——测试人员设计测试用例，在规定时间内找出许多缺陷；另一个测试人员加入后会设计出另一些测试用例，又找出一些缺陷；最后采用模拟用户行为的随机测试方法，又能发现些缺陷。因此人们会产生“我怎样才能不遗漏这些 bug 呢”的想法。

2. 如果在一个产品中我们没有使用模拟用户行为的随机测试方法，那这些应该被发现的缺陷是不是隐藏的并且是未被发现？退一步说，如果我们认为某个产品不需要经历随机测试就能发布，那我们应该承担这些缺陷被用户发现的后果。如果我们认为某个产品需要经历随机测试才能发布，则意味着增加了质量成本，并可能延期交付给用户。

当然，也有可能我们采用随机测试，而没有发现新的 bug，那么此时人们会不会想随机测试是在浪费成本呢。请大家继续讨论

A(iamredeye):

ongoing 的测试没有发现新的 bug 是一个信号，说明要调整策略：

- 1、不管是随机测试还是基于 test case 的测试，可能有效性不够。
- 2、可能客观上存在的 bug 不多了，不必继续测试浪费成本。

所以测试计划是需要动态调整的。

C (jane):

缺陷的存在似乎是不可磨灭的事实，综述上面的讨论，总结如下：

- 1、缺陷应该尽早被发现：比如说在评审需求说明书，在设计阶段，在编码阶段，甚至可以在测试人员编写测试用例时，在哪一步发现总比在最后整合时才发现来得好。
- 2、可以通过多种测试方式尽可能多的发现问题，当然缺陷永远存在，只要技术还在发展，人们还在前进，缺陷就会永远存在。其实这个时候除了在需求调研阶段尽可能的挖掘用户的原始需求之外，还应该在设计时花点精力。有的时候因为设计不当，不经意间会埋下不少隐患。设计问题带来的缺陷，其实对测试工作是个很大的挑战。个人总觉得，测试人员的素质是随着设计、开发人员的素质而随之波动。
- 3、关于发行后的缺陷，其实完全可以根据最初的需求描述。如果是因为需求描述不到位，那也没有关系，作为一种经验的积累。有的时候更多的是客户没有想到，所以有这类似的经验积累，可以为今后的用户做更多的引导工作。

注：如果大家还有更好的见解，可以上[www.step365.com](http://www.step365.com)搜索此贴，发表高见。

### 3 项目管理

#### ★项目管理从猴子说起

Q (cecilia):

从猴子说起

有这样一个笑话：一个旅客走进硅谷的一家宠物店，浏览展示的宠物。这时，走进一个顾客，对店主说：“我要买一只 C 猴。”店主点了点头，走到商店一头的兽笼边，抓出一只猴，递给顾客说：“总共 5000 美元。”顾客付完款，然后带走了他的猴子。

这位顾客非常惊讶，走到店主跟前说：“那只猴子也太贵了！”

店主说：“那只猴子能用 C 编程，非常快，代码紧凑高效，所以值那么多钱。”

这时，旅客看到了笼子中的另一只猴子，它标价 10000 美元。于是又问：“那只更贵了！它能做什么？”

店主回答：“哦，那是一只 C++ 猴，它会面向对象的编程，会用 Visual C++，还懂得一点 Java，是非常有用的。”

旅客又逛了一会儿，发现了第三只猴子，它独占一个笼子，脖子上的标价是 50000 美

元。旅客倒抽一口气，问道：“那只猴子比其他所有猴子加起来都贵！它究竟能做什么？”

店主说：“我们也不知道它究竟能做什么，不过它是做项目顾问出身的。”

虽然这只是一个笑话，但是有一点是可以肯定的，项目管理是非常重要的，而项目管理的人才又是极为缺乏的。在软件工业发达的国家，大家多少都知道点软件工程规划的重要性。在我们身边的台湾、印度、日本，都不乏因实施软件工程而成功的软件团体，更不用说身为软件大国的美国，已经从较低级的软件实现摆脱出来，进入了设计和营销的境界。

C (jane):

故事其实很简单，很直白的反应出目前软件行业中技术性人才的身价，顾名思义，项目管理人员的身价是相当高。其实也不光是 IT 行业有这个现象，整个社会最缺的两类人：一种就是管理人员，还有一种就是市场人员。（文中都提到了）

就从 IT 行业的管理岗位来说，TA 的专业技能相当广，不然如何去分配，如何去应对千变万化的风险。此岗位讲究的是一个人的综合素质。一个人如果要在开发、或测试上达到一定高度，这不难；但如果一个人要在很多方面都达到一定高度，这就不易。这更多的让我想到 SQA 这个角色，其实这就有点项目顾问的感觉，所以这是一份不简单的活。趁此机会也给所有 SQA 人员加油鼓劲，祝愿能早日成为真正的 SQA!!

### ★项目经理=什么

Q (老生):

估计来这论坛混的老师都是项目经理级别的，看看大家各自在项目管理中对于我所提到的几点大家如何处理……

- 1、客户对于项目所出现问题的压力大家如何面对；
- 2、公司领导对项目的管理完全依赖项目经理，而项目经理只有干事的份，无说话的份，怎么去改善这种情况？
- 3、项目组组员整体水平偏低，未真正有可以辅助项目经理的组员，全部都需要一点点的培养，这是上头给分配的团队成员，关于团队结构的问题，如何跟公司索要资源，关键是提了要求，上头老大还是按照他们的想法给分配人员。。

A (stepiv):

- 1、项目出现的问题，有多方面的原因，比如：项目技术实现上的问题、客户关系的问题、销售谈判的问题……等等。这些问题，职责不一定全是项目经理，这个看不同的公

司的情况及组织结构了。所以，这个问题太大，无法界定。

2、这个问题并不见得，项目经理可以做事，也可以说话的。既然叫项目经理，肯定是为整个项目负责，资源/进度/风险/成本/质量这些控制，如果都说不上话，那这个项目经理充其量仅仅是技术经验丰富者（相对来说）。所以，一般情况下，在项目启动时，就要跟公司签定合同的（相当于军令状），公司提供该支持的资源（软件、硬件、人员、资金等），其余的控制，就是看项目经理的了。总之，先弄清楚项目经理是什么？

3、索要资源，可以有两种方式：一种是正当的方式，即通过数据分析后，说明要增加的资源及理由，通过数据说话；另一种是不正当的方式，即故意让项目出状况，到时自然会顾及你这边的需求。

当然了，这个问题普遍存在，高层说给人，其实给一些经验不足或者直接就是应届生，这就很头痛，一方面还要培训这些人员，另一方面还要赶进度，真 TMD 搞死人，项目经理的时间全花在沟通等这些事情上去了。

其实，话说回来，BOSS 明摆着让这个项目拖进度，那就拖吧。

**A (lee\_huo):**

现在大多数企业还是管理制度比较森严的。有老总----有总监----有部门经理----然后是项目经理，其实这种制度就很大程度上扼杀的 PM 的自主管理项目的的能力。这个有点像刚刚看到的新闻，关于“杜伊”作为国奥主教练和中国足协的关系了。想按照自己的方式做事比较难。扯远了。

说回来，这些和你公司对项目经理的定位有关。如果把项目经理定位准确了，就会规避一些问题的产生。不过看到你的描述，估计你公司的项目经理都是有些技术能力，经验在 2 年左右的人吧，这样你的高层对这些人不放心导致的，都想插手管理，然后项目经理做事都得请示，导致项目经理没有什么权利只是义务了。

建议：重新定位项目经理的责权利；

项目的出了问题，不一定全是项目经理的责任，可以考虑一下公司有没有提供必要的支持；

有没有组织级标准过程、流程、指南、模板等提供给项目经理；如果没有提供这些项目出状况倒是必然的了。

**A (steplv):**

其实，我觉得这是把“技术经理”与“项目经理”的概念给混淆了。

老生提到的所谓“项目经理”，我在第一个回帖中也提到了，基本上偏技术型的人，跟

项目经理的概念差距很大。

建议：查看一下 PMP 中界定的项目经理是什么角色，或者，了解一下 PMI 的一些最新说明。

建议有志于往 PM 方向发展的朋友，看看这个：

PMI 定义的项目经理道德与行为规范。地址：

<http://www.step365.com/bbs/thread-460-1-2.html>

**A (lee\_huo):**

同意 steplv 的观点：弄清楚技术经理还是项目经理；

**A (老生):**

引用：

原帖由 steplv 于 2008-8-29 23:23 发表

3、索要资源，可以有两种方式：一种是正当的方式，即通过数据分析后，说明要增加的资源及理由，通过数据说话；另一种是不正当的方式，即故意让项目出状况，到时自然会顾及你这边的需求。

当然了，这个问题普遍存在，高层说给人，其实给一些经验不足或者直接就是应届生，这就很头痛，一方面还要培训这些人员，另一方面还要赶进度，真 TMD 搞死人，项目经理的时间全花在沟通等这些事情上去了。

非常感谢斑竹的答复，基本上你第 3 点的答复已经为我解惑了。

我想大家都会有向上头申请资源时会不太顺利的时候，换位思考一下，上头也要管理，不可能项目经理凭空要什么就给什么，所以，我们要做的是给上头真实的数据说明索要资源的理由。这不就正是项目经理所需要具备的技能吗。

对于其他老师的回复在这我也谢了，我想不管那个项目都会有这样、那样的问题存在，如何在现有资源的环境下把项目管理好，这就是考验项目经理的能力了。

PS:

关于项目经理定位的问题，我想这得看项目大小了，项目规模在 7—8 人的话，不可能会有什么架构师或者技术经理人员存在，所以，项目经理也就是带此规模的项目是最累的……

**A (steplv):**

通常情况下项目经理会兼作好多工作的。不过即使没有构架师、系统设计响应的人员，也需要把这个角色列出来，然后项目经理兼任。这个会从侧面反映出公司人员角色的缺失，



必要时会给高层经理以提示的作用。

C (jane):

上面有人解答挺到位了，在这里我还想再补充点个人的看法。针对第二个问题，我想说，公司应该设立一个独立的部门来收集项目中的一些问题，可以为项目组与上层领导的沟通搭建一座桥梁，这也是为了公司发展考虑。当然在设立这个部门前，肯定得用 **Steply** 的方式，用数据说话。当然说到底也就是个组织架构的问题，若想改变组织架构，也需要在一定程度上让高层意识到目前的架构在实际操作中是有问题的。如果你们能收集点信息，向领导反应，我想领导也会相当关注的。

### ★ 如何进行质量分析

Q (xixiaojing666):

今天领导让我们作个表，能展示所有项目的项目进展，在这个表中需要有目前所有进行的项目。每个项目要写出项目成员、进度情况等基本的监控项。

不知道各位在项目的监控方面是如何做的？欢迎大家讨论，并给我帮助！

A(lily\_014):

因为没有发现其它的好的工具，因此目前可以采用 **excel** 表格来做这件事情。一点建议：

比如：项目名称、项目经理、项目类型、项目当前状态等等根据你的目标需求来设置这些项，记录的这些都是相应的数据，也正是展现给领导看的，领导只关心这些数据，并不会关注到每一个具体的问题，因为有问题需要他们协调时，你自然会报上来。他们只关心这些数据是否正常，如果出现异常了有没有解决（这些数据可以根据你们的允许偏差范围来设定一定的条件，如进度延期了少你就设定为一个显目的颜色标注，算是一个提醒）。为了表格观看方便，美观，项目组成员的分布和归属可以另起一个 **sheet** 表，在同一个工作簿内。

A(xixiaojing666):

领导具体关心哪些数据呢？详细的和我说说

A(lily\_014):

呵呵。。至于领导关心什么数据。。那要和你领导详细说说了哈……

不同层面的领导关注的可能不同。对于高层领导可能更多的关注点是，进度、成本以及回

款吧：)

**A(xixiaojing666):**

谢谢呀，我们领导好象只提到了进度，成本目前我们无法统计。

目前我们初步讨论出了一个统计项目人员状态和项目进度的表格，不过总是觉得进度表现的不是很好，目前正在继续酝酿中！

**A(step365he):**

项目里程碑和节点的进度是否有延期；

在里程碑和节点到来前是否能给予提醒或者重视；

出现延期的里程碑和节点，是否会对接下来的里程碑和节点造成影响延期的，是否需要投入更多资源风险我能否知道，风险的后果我能否估计……这些是领导们想知道的。

**C (jane):**

一般项目的监控是根据项目启动后制定的项目计划来的。不管用什么工具（比如：Project）来制定计划，需保存计划评审后的基准值。根据计划中的里程碑点，SQA 可以介入跟踪，通过对阶段工作的审计，领导关注最普遍的应该是能否按时完工，所以在每个里程碑将审计内容反馈给相关领导。

#### 4 QA/EPG 专区

### ★ 项目中代码评审没执行——谁的责任？

**Q (jane):**

进新公司，第二周，开始正式跟踪一项目。

从计划评审开始加入，那次评审会，基本上处于业务不熟悉的状态。所以在听评审会时也不是很清晰，当时的计划有什么问题。不过前SQA人员倒是在会议中提了不少点。因为由专门的记录人员，所以也没在意记录的内容。

此项目的有个特点就是，项目中附带了一个需求，而且此需求由专人负责开发，发布。当时SQA就提醒项目PM将这部分工作放在计划中。事后，PM也将此任务点放在计划中。

结果，在开发计划评审会议时，突然发现独立的需求没有进行SQL语句的评审，而这一步是相当关键，在流程中是必不可少的。更严重的是，此需求的发布没有预先申请，导致相关的工作没办法按常规配合起来，直接影响到项目发布进度。



此问题还得没有结束，下周一对项目 PM 和另一独立需求的负责人调查原因。

深思一下，其实我的责任相当大，没有监督好项目流程，没有保证流程的质量。

**A (lee\_huo):**

责任肯定有；作为关键评审没有被执行，是很可怕的事情。不过这种评审按照你们的规定是否可以不采用评审会的形式呢？项目组或 PM 是不是在组内进行评审了呢？如果经过组内评审，没有评审记录或没有通知 QA 这个问题就比你描述的问题的严重性轻多了。

如果组内也没有做，正式的评审也没做，而且没有被 QA 审计发现，那就是整个项目组、PM 和 QA 的事。

不过发生这个问题还是值得深度挖掘的。为什么不执行评审？qa 为什么没有及时发现？计划是否有变更而 QA 不知道？或者延迟评审等？或者 PM 或项目组对过程本身有不理解或有意见。发生问题就是为组织提供改进的最佳时机。分析分析原因吧。可能有意外收获！

**A (jane):**

嗯，今天又跟相关人员沟通了一下，问题有了新的发现。主要问题出在需求提出的人员将两部分需求分开执行，走不同的流程。结果两位 PM 均忽视了审核的过程。只关注最终功能实现的问题。看来以后定义项目的范围还得有待加强。

**A (xixiaojing666):**

我觉得 QA 有一定的责任，首先需要监督是否将这个评审活动纳入到项目计划中，其次，要监督项目计划中的每个活动是否按照组织规定执行。

QA 在每周的检查当中，应该发现本周的不符合项问题。

所以 QA 在做每周检查时，不要流于形式，要认真检查

**C (jane) :**

“不要流于形式”这话不错，制定流程的同时，还得学会理解流程，应用流程。

### ★ SQA 如何收集度量数据，建立有效度量体系？

**Q (cecilia):**

目前对于度量这一块还是比较迷茫，不知道应该怎么做。

在项目开始立项后，项目经理都会编制《项目计划》，然后通过产品经理和项目组成员的评审，之后再纳入配置库，进行发布。然后由项目经理对整个项目进行监控，SQA 督

促项目组成员更新 ProjectServer 上的数据,可这里就有一个问题,大家目前还没有这样一种习惯,每天主动去更新数据,经常是要到项目经理或 SQA 人员去催促才会去填写相关的信息,这些数据也是有水份的,对于度量来说就是无效,尤其是实际开始及实际完成时间那一块,因为没有及时更新,导致后来忘记。

到编码阶段,因需求变更、方案改变等各种因素,导致项目计划没有及时变更,里程碑会议也未及时召开,在这个阶段,度量就无法再做了,不知该从哪里去收集数据。之后就只有周例会,每周都会有列计划任务,如果未完成将会有相应的原因分析、解决办法和具体解决时间等。

基于以上两种情况应如何收集度量数据呢?主要有哪些方式方法呢?希望大家指导。

**A (xixiaojing666)**

我觉得你存在两个问题,所以导致度量无法进行

- 1, 不及时登记实际的数据
2. 计划发生大的变更, 不去更新计划

这两个问题 QA 要及时催促了,等过一段时间,就会养成习惯的,我们开始的时候,计划等所有工作都是 QA 协助项目经理做的. 经过了一年多,基本上 QA 只要督促,不需要协助了,呵呵,耐心点了,这种状态就会好点了

**A (cecilia):**

谢谢晓静的回答。

看来只能是多多催促,督促项目经理完成了。如果在多次提醒项目经理还无果的话,就只能是上报给高层了。要做一个合格的 QA 了,除了要掌握相应的知识的,耐性也是非常需要的。可能在多次催促无果的情况下,会产生厌烦等情绪,这个时候就得好好调节自己的心态,不可有抱怨心理,应及时地去发现问题、分析问题及解决问题。(给出解决方案)

其他人还有什么意见或建议呢?最近大家都不怎么发言了。

**A (lee\_huo):**

要想做好度量就和组织的目标挂上钩,组织需要、高层关心的、我们就收集,分析。这样方便与项目组沟通。记住要与公司的商业目标相一致。

**A (psli1983):**

同意!

度量必须和组织经营目标相关度量采用意义,同时,通过度量数据,项目经理能一眼就能看出项目信息的,这些度量项,项目组也一般愿意度量的。

所以,定义度量指标很重要哦。至于:

- 1.不及时登记实际的数据;
- 2.计划发生大的变更,不去更新计划;

应该是公司度量规程里面规定的了吧,没有按照过程执行,只有 QA 及时去和项目经理沟通,讲道理(讲述不及时更新的利弊),定期提醒,定期跟踪了哟。

**A (iamredeye):**

数据收集除了要能给老板们带来价值,还得让被要求输入数据的项目成员看到这些额外的工作带来价值.至少让他们看到数据被怎样使用.  
否则输入数据不会形成习惯的;被强迫成习惯数据也不会准确的.

**A (cecilia):**

确实是组织本身应先定义一个度量目标,如何收集这些数据,有了这些数据应如何去计算等都应有个定义,需要这些数据的目的是为什么。如此项目组成员才会配合 填写,当然 QA 也需要不断的催促,将组织的目标传达给各项目组成员,由其认真填写相关数据,以作为后期其他项目的经验数据等。

其实关键的问题还是如何让项目组成员对所有这些规范及流程等重视,因现在都觉得是组织逼迫其做的,本身都没有习惯去这么做,QA 如果一味地去催促,会让项目组成员觉得 QA 根本就是在找茬,会对其产生敌对情绪。

如此一来,QA 要开展工作就存在很大的阻力。

希望大家能多多提意见。或许很多的建议都很好说,可到真正实施的时候就去有困难

## 5 CM 专区

### ★ 配置管理带来的好处

**Q (sungubbi):**

配置管理给我们带来一些什么好处?

今天开发人员在做项目维护的时候,导入了 bug,于是我们通过 CVS 将导入 bug 的版本找出来,解决了 bug。

呵呵,这就是配置管理给我们带来的好处之一吧。

补充一下,通过将近 1 年的配置管理,我们的代码管理初见成效,最近两年的代码都得到了统一的管理和备份,以前那种只有开发人员手里有代码,公司没有代码的事件不存在

了。

也许这只是初级阶段,不过还是有成绩的。

各位能谈谈自己的切身感受嘛?

**A(cecilia)**

依据目前来讲,主要就是对工作成果的一种有效保护。

### ★ 备份记录的问题探讨

**Q (sungubbi):**

我们目前的备份大部分都是自动执行,但是 CM 需要进行定期检查,我们一般一周做一次纸制记录。

记录备份的执行情况、备份内容、备份检查人、检查日期等等。目前已经登记了很多很多,我感觉这个任务很烦,也没有人看,总是有不想记录的感觉,但是如果不记录,又不能说明你已经检查过。但是每周重复的记录,觉得也没有什么意思,不知道大家是怎么做的?有什么更好的办法?既能保证备份的有效性,还能去掉过多的烦复活动?(备份项目开发库\受控库\产品库以及组织等数据)

**A(iamredeye)**

我觉得也没什么意思. 目的或价值在哪里?这样的方式能达到预期的目的吗?人人说起 CMM, 都提到形式主义, 虽然不完全正确, 还是有原因的.

## 第三部分：会员原创

### 1 CMMI 方面

#### ★ 实施 CMMI 时必须解决的认识问题

作者: jaedonger

在基于 CMMI 实施软件过程改善时,有些根本的思想认识问题解决不了,往往会使实施的周期比较长,效果不好,甚至导致过程改善的失败或中止。软件企业的高层领导、企业的过程改进主管、项目经理及一般的开发人员都需要对这些问题统一认识,在此基础上才能消除各方面的阻力,把握好过程改善的方向,控制好过程改善的进度。CMMI 不是万能的,只有 CMMI 是不行的,还要技术(开发方法、工具)、人员二个要素一起改善。

在软件工程发展的历史进程中,人们为了解决软件危机,尝试了采用诸如形式化描述语言、结构化开发方法、CASE 工具、构件化开发方法等等各种解决方案,但是效果并不那么显著, SEI 提出了软件过程能力成熟度模型集成(CMMI)基于过程的角度来解决软件/系统危机。那么是否实施了 CMMI,软件企业的开发能力就一定能提高,一定能带来经济效益呢?答案是否定的。如果企业里要带来经济效益必须要结合软件过程、工具、开发方法、人员等多种因素一起提高,才能保证带来经济效益,因为人员、技术和过程是支撑软件开发平台的三条腿,少了那一条都不行。大家也都知道木桶原理,一个木桶可存储水的最大容量是由最短的那根木头决定的。在企业的开发能力中过程、技术(含工具、方法)、人员都是主要的因子,都需要全面提高,只关注一个方面,而忽略了其他方面,都是有害的。

在开始实施 CMMI 时,最容易犯的一个错误就是“唯管理论”或孤立地只抓过程改善,忽略了开发技术与人员的提高,过分强调管理的作用,实施了半年或一年后,发现企业的生产能力并没有得到明显的改善,这时反对的声音就会成为主流,过程改善就难以继续进行了。有的企业采用面向对象的开发方法进行软件开发,但是企业内并没有对面向对象技术真正了解的专家,虽然也采用 RUP 过程、也采用 ROSE 等开发工具,但是仅仅是形似,没有做到真正的 OO 方法,没有得到 OO 方法的精髓,这种问题仅仅依靠过程改善是无法解决的。还有的企业开发人员的积极性很差,工作热情很低,企业的激励机制没有起到很好的作用,这种问题也是依靠 CMMI 无法解决的。

1. 管理就是预防,管理的作用是隐性的,不都是立竿见影的,大家要有耐心。

在实施 CMMI 时,企业的管理层在开始时往往会对过程改善期望值太高,希望短时间内效果显著,正如上面所述,效果显著与否不是由一个方面的要素决定的,需要多个因素共同改善。而管理的最大作用是预防,防患于未然。

任何的管理的改善都是符合 J 曲线的,即在改善的初期企业的运行效率可能会下降,甚至可能会出现一些混乱的局面,不过渡过了这段时间就会看到效果。所以在改善的初期大家要有这个思想准备,要有耐心。

## 2. 坚持活学活用,以我为主

机械照搬 CMMI 的条文是在实施 CMMI 时常犯的错误。在国内的软件企业中,都是从作坊式的软件组织逐步发展过来的,也没有经过系统工程化阶段,甚至也没有什么标准、规范。真正超过 10 年软件/系统工程经验的人员更是比较少的,加上大家不愿从事管理,因而有的企业所组建 EPG 的人员中可能在工程经验方面是有欠缺的,又没有真正的有实践经验的专家进行指导,所以对 CMMI 的理解就不可能一下就很深刻,不敢裁剪 CMMI,容易机械照搬 CMMI 条文,其实这恰恰违背了 CMMI 的精神,CMMI 是系统工程经验的集大成,是从实践中总结出来,用以指导实践的,CMMI 本身也在更新版本,不断完善。每个企业都有自己的特点,就象微软的 MSF,那是微软自己内部的管理过程标准,是微软的产品开发经验总结,有些内容是 CMMI 中没有的,完全可以借鉴过来使用,所以只要可以提高企业自己的软件管理水平,就应该大胆地来尝试。

在推行 CMMI 时,所以遇到的阻力,很大程度上是由于照搬 CMMI 的条文,不切合项目组的实际,没有具体情况具体分析。实际上,一线的管理人员、开发人员最了解实际。谁了解实际,谁有发言权。所以在制定 CMMI 规程标准时,尤其是在制定大家要执行的操作规程、模板和记录模版时,一定要得到执行者的认同,否则就容易造成执行和沟通的障碍,你硬要推,表面上看来似乎大伙也照规程做了,其实是表面文章,对改善没有实际帮助,以导致 SPI 工作受阻。

## 3. 要改良式不要革命式

以革命的方式来实施 CMMI,期望通过一场运动来解决过程能力的问题,一种可能是不懂 CMMI,不晓得管理的改进是循序渐进的,一种可能是明知故犯,期望在短期内通过 CMMI 评估,单纯追求市场上的轰动效应。有的企业在短时间内虽通过了 CMMI 几级评估,我想恐怕由于没有实效而得不到大家的认同而难以将这种"水平"持续下去。一个企业引入 CMMI 之后会从本质上影响企业的文化,改变大家的思想与做事方法。有人曾形象得将过程改进比喻为减肥,你是可以依靠减肥药在短时间内减轻体重,但如果不从根本上



改善饮食、生活、运动的习惯，那么体重将会在短时间内恢复到原来，我认为这个比喻是十分形象的。

## ★ 研发过程如何进行裁剪

作者：yuanjing

项目特点是裁剪依据和出发点。裁剪指南应包括以下内容：

明确可裁剪的对象：可裁剪对象确定了裁剪的范围，可裁剪对象不仅限于过程元素和活动，还包括标准、方法和工具、输出的工作产品及模板等。

确定裁剪所考虑的要素：对于某个裁剪对象，其范围、频度、正式度等都是裁剪要素。如，对于已有类似开发经验的项目，可以适当减少过程培训、业务培训等活动；对于开发周期较短的项目，可以适当合并一些评审活动，如概要设计和详细设计评审合并进行。

项目在进行裁剪时，由于裁剪指南很难枚举所有的裁剪情况，因此有时还是需要项目经理和 QA 依据经验进行判断和决定，这时，最根本的依据就是项目的质量要求和对风险的考虑。首先要分析如果一旦裁剪掉某些活动，是否会给项目带来风险，带来多大的风险，以及是否影响项目质量目标的达成。然后综合考虑后才能决定是否裁剪，如何裁剪。另一方面，企业建立标准过程的目的是为了“为了规范而规范”，而是为了提高过程和技术的双重用。

因此，如果项目在裁剪时有很大的灵活度，每个项目定义的过程都很随意或者项目过程之间相似的内容很少，那么重用的目的就很难实现了。所以，规范度和灵活度是项目裁剪时需要平衡的另外两个要素。

概括之，过程裁剪的原则是：质量与风险并重，规范与灵活的平衡。

### 一、企业在应用过程裁剪时的常见问题

不论企业实施了 ISO9001、CMMI、六西格玛，或是其它任何类型的质量管理体系，通常都会形成完整的公司级标准过程体系。但当项目经理需要在项目中使用这个已定义好的过程体系文件时，面对厚厚的过程文件往往无从下手，心中也充满疑虑：

1. 我的项目开发周期只有 3 个月，团队 4、5 个人，难道要完全按照公司定义的标准过程执行吗？如果必须执行所有的过程和子过程，生成所有要求的技术和管理文档，那项目的开发周期恐怕不是 3 个月，而是 4、5 个月了。那我的项目还能成功吗？

2. 我听说过“裁剪”这个词，不过到底是“裁剪”还是“裁减”，我还没有弄明白。

即便弄明白了应该是“裁剪”，是 Tailoring，而非“裁减”，可具体该怎么操作？我可以随心所欲将自己认为不必要的或者很费时费事的过程裁剪掉吗？

3. 如果公司有 QA，也有《裁剪指南》，那就好办了，我可以在 QA 的帮助下使用《裁剪指南》裁剪得到项目的过程，执行就是了。但如果公司没有 QA 的角色，我就只能自己进行裁剪了。可是，裁减的结果需要有人批准吗？

在这里，我们假定完整的公司级标准过程体系是包括了企业的方针、过程、指南、模板和表单等一整套的体系。那么，项目经理该如何是好？

## 二、过程裁剪的目的和作用

建立裁剪指南的目的是用来指导项目对组织标准过程（Organizational Standard Process, OSP）进行裁剪，以形成符合项目特点的项目定义过程（Process Defined Process, PDP）。

组织标准过程是在企业的层面上描述的，它包括了开发一个完整产品/项目的全过程，以及相应的支撑过程，它是一个企业运作的过程的全集。因此，每个特定的项目都可能无法直接使用组织标准过程。比如，组织标准过程描述了开发一个系统级产品的完整过程，开发过程中包括了软件、硬件、结构、工业设计等开发过程。而某个特定项目仅仅包括纯软件的开发工作，在这种情况下，该项目无法也不应该盲目遵照执行完整的过程。或者，某个特定项目，项目的成功标准是按时交付，而客户要求的项目交付期特别短。为了达成这个目标，项目也不得不对过程进行裁剪以满足客户的需要。裁剪指南就是来帮助项目裁剪组织标准过程，以形成项目定义过程，使用项目定义过程来管理项目，实现项目的目标。

裁剪指南能确保所有项目在定义项目特定的工程活动、需求开发和管理、计划、监控、测量分析、配置管理、质量保证过程时有一个共同基础。裁剪指南主要可在以下方面指导项目：

1. 选择适当的生命周期（是组织标准过程中的一部分），由于各种生命周期模型在软件工程领域已经有深入的研究，业界对于瀑布模型、迭代模型、增量模型、螺旋模型的使用场合等也基本达成了共识。因此，项目只需要将项目的实际特点与生命周期模型的应用场合相匹配，选择合适的生命周期类型即可。

2. 剪裁组织标准过程和所选择的软件生命周期，使之符合项目的具体特点。

## 三、如何进行过程裁剪

### 1. 裁剪的原则

本文中多次提到“项目特点”一词，项目特点包括了：①项目规模，如大、中、小等，通常可以使用功能点（Function Point）或 KLOC（千行代码）、单板数等单位进行度量；②



项目类型，如开发、维护、功能增强等；③项目技术复杂度；④项目周期；⑤产品种类等要素。项目特点是裁剪依据和出发点。裁剪指南应包括以下的内容：

(1) . 明确可裁剪的对象。可裁剪对象确定了裁剪的范围，可裁剪对象不仅限于过程元素和活动，还包括标准、方法和工具、输出的工作产品及模板等。

(2) . 确定裁剪所考虑的要素。对于某个裁剪对象，其范围、频度、正式度等都是裁剪要素。如，对于已有类似开发经验的项目，可以适当减少过程培训、业务培训等活动；对于开发周期较短的项目，可以适当合并一些评审活动，如概要设计和详细设计评审合并进行。

项目在进行裁剪时，由于裁剪指南很难枚举所有的裁剪情况，因此有时还是需要项目经理和 QA 依据经验进行判断和决定，这时，最根本的依据就是项目的质量要求和对风险的考虑。首先要分析如果一旦裁剪掉某些活动，是否会给项目带来风险，带来多大的风险，以及是否影响项目质量目标的达成。然后综合考虑后才能决定是否裁剪，如何裁剪。

另一方面，企业建立标准过程的目的是为了“为了规范而规范”，而是为了提高过程和技术的重用。因此，如果项目在裁剪时有很大的灵活度，每个项目定义的过程都很随意或者项目过程之间相似的内容很少，那么重用的目的就很难实现了。所以，规范度和灵活度是项目裁剪时需要平衡的另外两个要素。

概括之，过程裁剪的原则是：质量与风险并重，规范与灵活的平衡。

## 2. 裁剪的过程

裁剪过程有这样一些主要活动：

(1) . 根据组织标准过程和裁剪指南，进行过程裁剪，以符合项目特征。项目经理在 QA 的协助下完成该工作。

(2) . 记录裁剪的理由，将裁剪的结果整理成项目定义过程文档。

(3) . EPG（工程过程组）审核裁剪理由和项目定义过程，并批准。审核的检查点主要包括：是否与组织标准过程一致，是否符合本项目的特点，是否记录了充分的裁剪理由。如果审核不通过，则重新进行过程裁剪，或进行修改。

(4) . 使用项目定义过程就是要基于项目定义过程制定项目计划，根据计划监控项目的实施。

## 3. 应避免的误区

### (1) “裁剪”而非“裁减”

常常见到企业的过程体系中赫然存在一份《裁减指南》，员工也往往认为裁剪就是大刀

阔斧地“减少”完整的过程要求。如果项目时间紧、缺乏资源，就可以这么做。这是一个认识的误区。

所谓“裁剪”就是量体裁衣，根据项目特点量身定做最适合项目的过程，以期项目用最经济的过程实现质量目标。对于一个开发周期超过 1 年的系统级产品的开发，公司定义的四大决策评审点：概念决策、计划决策、可获得性决策和生命周期决策，以及六大技术评审点，技术评审 1 至 6，可能“一个都不能少”。而对于一个快速定制开发的项目而言，很可能只需要将概念和计划决策合并为一个决策评审点，某些技术评审点也可以合并。

## (2) 直接用项目定义过程来管理项目

有些项目经理认为裁剪得到了项目定义过程，然后就可以开始项目的具体工作了。但项目定义过程并非项目计划，更不能替代项目计划。项目经理应基于项目定义过程制定项目的 WBS（工作分解结构），以 WBS 为基础进行工作量、规模和进度估算，制定项目进度表和完整的项目计划。后续工作要以项目计划为基础监控项目的实施。

在过程中，项目还要记录、收集和分析实施中的度量数据，用于监控项目。一些常见的问题、风险和经验教训总结更应该提升到组织层面进行统一管理和协调。从而不断改进组织标准过程，形成闭环。

## (3) 不允许裁剪过程，或者裁剪有很大的灵活度

有些企业在刚刚建立过程体系时，由于很难立即制定一份完善的裁剪指南，所以干脆一刀切，不允许裁剪过程。但这样硬性规定的结果是一些维护型、功能增强型的项目要么就是在搞不清状况的情况下照着完整的过程执行，生成很多文档，也延误了开发周期，降低了效率；要么干脆拒绝执行过程，仍然按照过去的工作方式开发。显然，这就违背了建立过程体系的初衷。

另一个极端是企业允许过程裁剪有很大的灵活度，却没有设定一些原则。这样的结果往往是项目随心所欲地裁剪过程，最终项目形成的过程资产的可重用性非常低。

针对规范性和灵活度的平衡，很多企业倡导“先僵化，后优化”的原则，在过程建立的初期尽量削足适履，等到积累了一些经验后再优化过程，形成更易操作的裁剪指南。

## 2 TEST 方面

### ★ 测试方案和测试计划的区别

作者: lee\_huo

### 一、测试计划:

对测试全过程的组织、资源、原则等进行规定和约束,并制订测试全过程各个阶段的任务以及时间进度安排,提出对各项任务的评估、风险分析和需求管理。

### 二、测试方案:

描述需要测试的特性、测试的方法、测试环境的规划、测试工具的设计和选择、测试用例的设计方法、测试代码的设计方案。

三、测试计划是组织管理层面的文件,从组织管理的角度对一次测试活动进行规划。

四、测试方案是技术层面的文档,从技术的角度度一次测试活动进行规划。

### 五、测试计划要明确的内容:

#### 1、明确测试组织的组织形式

1 测试组织和其他部门关系,责任划分。

2 测试组织内的机构和责任安排。

2、明确测试的测试对象(明确测试项,用于后面划分任务,估计工作量等)

#### 3、完成测试的需求跟踪

#### 4、明确测试中需要遵守的原则

1 测试通过/失败标准

2 测试挂起和回复的必要条件

#### 5、明确测试工作任务分配是测试计划的核心

1、进行测试任务划分

2、进行测试工作量估计

3、人员资源和物资源分配

4、明确任务的时间和进度安排

5、风险的估计和规避措施

6、明确测试结束后应交付的测试工作产品

### 六、测试方案的具体内容:

#### 1、明确策略

2、细化测试特性(形成测试子项)

3、测试用例的规划

4、测试环境的规划

5、自动化测试框架的设计

6、测试工具的设计和选择

七、测试方案需要在测试计划的指导下进行，测试计划提出“做啥”，而测试方案明确“咋做”。

### 3 RD&RM 方面

#### ★需求编写的几点经验之谈

作者: yuanjing

近年来,“需求管理”正成为中国当前工程应用和商业热域的热点。目前,有关需求管理的实践大量应用于软件开发工程等领域,软件开发团队在开始一个新的项目之前,会通过详细的用户需求调研准确捕获了用户需求并汇总分析后,再进行下一步的设计与实施工作,避免因未能正确识别用户的真正需求而导致不断返工和工作成本增加。对于从事软件工程的程序员们来说,在进行项目开发之前创建和管理良好的需求是非常重要的第一步,同时也是一项挑战。需求表述不当可带来重大影响,如耗时返工、延期交付及预算超支,严重的还可造成业务违规。因此,开发团队需要首先有效定义和管理需求,才能确保在保证进度和控制预算的同时,产品能够满足用户所需。本文旨在阐述良好需求描述的特征,并介绍有助于更好地编写软件工程需求说明文档的几点经验,以帮助软件开发团队能够更快更好地取得投资收益。

##### 1. 高质量需求的特征

首先的问题是,何为良好的需求?一般而言,一项编写良好的需求描述,应该包含以下特征:

良好需求的特征	含义
正确(Correct)	技术可行,内容合法
完整(Complete)	能够表达一个完整的想法
清晰(Clear)	不模棱两可,不易被误导
一致性(Consistent)	不与其它需求相冲突
可验证性(Verifiable)	可验证系统能够满足用户需要
可追踪性(Traceable)	可唯一识别并进行跟踪

可行性(Feasible)	可在预期成本和计划进度内完成
模块化(Modular)	可单独变更而不会造成较大影响
独立于设计(Design-independent)	不包括项目设计和实现的细节、计划信息等

## 2. 提高需求编写质量的十佳经验

在明确了何为良好的需求之后，以下介绍几点可以帮助开发团队编写出更好的需求描述的方法，加速软件工程投资回报率。

### 经验 1：将需求结构化(Structuring)

每一项需求既不能被重复描述也不能被遗漏，诀窍之一是将需求结构化。需求组织应具有良好的结构，以增进理解，同时避免出现重复和忽略的情况。同时，须具备对需求的向上和向下的追溯能力之后，团队才能够评估需求的覆盖范围。结构化组织需求是控制和改善需求质量的第一步。

### 经验 2：重视非功能性需求(Constraints)

对于编写需求说明书而言，涉及法规遵从和提高软件系统质量的非功能性需求（又称约束条件，Constraints）同样重要，它们通常包括软件的性能、界面和可维护性等方面。编写良好需求应包含对约束条件的覆盖，原因是一旦如下领域（例如，性能、可靠性和易用性等）在开发完成后出现缺陷，通常都无法在系统中对其进行重新设计。因此，在项目初期将所有类型的非功能性需求考虑在内，可帮助开发团队大幅提高项目成功的几率。

### 经验 3：将需求可视化(Visualization)

大多数需求分析人员发现建模有助于直观化文字形式的需求。无论是在白板上绘图、使用 Microsoft PowerPoint 演示工具，还是仅仅在脑海中构建一个模型，都可视为一种建模方法。以上这种图型化的文档应与文字形式的需求描述一起统一管理，以确保一致性、可跟踪性和变更控制能力。可视化需求建模提供了一种与客户及最终用户沟通的简单而有效的方法，通过该方法可较容易地掌握客户和最终用户的需求。此外，图型化还有助于阐明需求，增进软件项目所有相关人员之间的沟通与协作。

### 经验 4：使需求具备可测试性(Testable)

产生良好需求的另一种行之有效的方法，就是从初期就确保每个需求具备明确的可验证性，这种做法不仅有助于为项目后续阶段做好准备，还可以帮助编写者保持正确的思路。对于非功能性需求此规则也同样适用，例如，对于“软件必须具有高可用性”这种表述的需求我们无法进行测试，而改写为明确的“普通用户应能够在 3 分钟内生成一个报告”就使该需求具备了可测试性。

### 经验 5: 管理好需求变更

大多数软件工程项目中, 来自用户的需求经常会有变化。随着项目的进展, 开发团队要保持清醒的头脑、按照工程要求做出相应调整, 并响应不断变化的市场形势和客户需求。仅仅编写出完美的首版需求描述是不够的, 如果未能对需求的变更过程进行恰当管理, 那么控制不善的变更便可能导致系统和软件功能缺失、返工以及利润损失。开发团队应该实施可靠的、可重复的变更控制流程。

### 经验 6: 在客户需求和开发能力之间找到平衡

许多情况下, 较少的需求数量有助于产生更加优秀的需求描述。软件工程项目不可能实现既采纳和满足企业所有用户的需求、营销理念和商业计划, 同时还符合预算并能按期交付。项目经理必须找到客户需求和开发能力之间的平衡点, 确定可为客户带来最大价值, 并帮助企业提升创新能力的那些需求, 而不是一味地试图满足用户所有需求。

### 经验 7: 正确的重用以往优秀需求

当之前项目的已编写的良好需求适用于当前情况时, 不要单纯地将原有需求直接复制。重新使用以往需求的正确方法是继续维持两个需求之间的联系, 如通常打上 re-use 标记。此标记使分析人员能够随时查找到原始需求, 以检查需求分解分配等信息。通过灵活的方法重新用以往需求, 开发团队可以获得技能、经验和知识的共享。

### 经验 8: 建立范例知识库( Knowledge Database)

提高需求质量的另一有效途径是建立范例知识库, 并参考其中的典型范例。知识库内容应该包括: 良好需求和文档的正、反面示例, 以往项目中可反映团队在特定领域内专门知识的良好(和不良)需求。为了使开发团队可以更好的参考, 知识库中的需求案例应具备明显的积极或消极意义, 而非中规中矩的。通过知识库示例开发团队可以参考以往的经验、吸取教训, 避免重蹈覆辙, 进而提高需求编写的质量、一致性和完整性。

编写好的需求说明是一个开发项目最为重要的活动之一, 优秀的需求描述可以改善并加速项目的投资回报。就好像“垃圾输入, 垃圾输出(garbage in, garbage out)”所表明的那样, 如果前期用户需求收集得不明确, 那么后期的开发过程注定生产错误的产品。开发团队可以通过经验提升需求编写质量。此外, 通过应用业界领先的需求管理工具, 可以优化项目开发的沟通和协作的过程, 提升软件项目过程质量。