

【软考达人】

# 软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



**微信扫一扫，立马获取**



**6W+ 免费题库**



**免费备考资料**

PC版题库: [ruankaodaren.com](http://ruankaodaren.com)

## 论软件过程改进

### [摘要]

本文以我近期参加的一个进销存的信息管理系统为背景，讨论一下软件过程的改进。近年来对软件产品质量要求日益提高，而且软件越来越复杂。为了提高软件质量，许多厂家都在改进自己的软件生产过程。改进软件产品的生产过程能够改进软件产品质量，是基于生产中的产出物（包括各种文档、代码等）能够保证质量，生产出的软件产品也就能够保证质量。此系统的用户为一家大型外企，它要把产品从国外运到国内出售，此系统就是负责管理进货、销售、库存各环节。另外，它还要跟用户国外工厂的系统有联系，本公司在软件项目管理有些问题，针对这些问题谈下体会。

### [正文]

这是一个进销存的系统，我在项目中担任系统分析和设计。用户为一大型外企，系统是有两部分，一部分是进销存的管理系统，一部分是接口。由于国外工厂是按订单生产的，所以这边需要把采购的定货信息及时发到国外工厂，以便工厂进行生产。国外工厂的系统与用户国内的 SAP 系统有接口，所以此系统是通过 SAP 系统与国外工厂的系统交互数据。SAP 是一种大型企业信息管理系统，用户的 SAP 系统还与用户其它的系统有接口，所以此系统设计时需要符合它们的公共规范。此系统还会从 SAP 接收共用的基础数据，有四种数据是发给 SAP 的，有五种数据是从 SAP 接收过来的，两个系统的通信是通过 wbi 的 MQ 来实现的。而 wbi 与 SAP 系统和此系统是通过文本来交互数据，在指定时间 SAP 将要传给本系统的数据生成文本放到指定目录。在检测到文本文件后，wbi 将会生成 MQ，MQ 发到本系统服务器上的 wbi，wbi 会把 MQ 转成文本文件，本系统将其数据读入到数据库。同样，本系统也会通过同样的方式把数据发到 SAP。

在进销存管理部分，业务人员会根据系统提供的库存，销售的信息，制定定购的数据，这些定购数据会在固定的日期（每周五）发到 SAP 系统，通过它传到工厂系统里。

本公司在软件项目管理存在一些问题，现说明其中两点：

#### 1、需求管理问题

需求管理一直是个重要问题，为用户开发的系统最终是给用户用的，如果做出的产品不是用户要的，那项目肯定就是失败的，在这个方面，公司以前的项目存在两个方面问题。

##### （1）项目成员之间对需求理解不一致

在做需求分析时，大家会将要求记录下来。由于某些原因，如：用户的需求会不断变化，有可能转变成以前舍去的需求，还有项目组成成员考虑事件的立足点不一样，这些都会造成理解的不一。虽然大家可以看文档，但这就存在版本冲突的问题。由于理解不一样，使得项目组之间时常出现争论，以致时间不必要的浪费。

举例如下：

在用户发给国外工厂的定单中有两种，一种是真实的定单，就是说用户需要定这么多货，不会再更改。另一种就是定未来的货。定单的发送是有周期的，如果某种产品定货周期为一周，那么就是这周要定下一周的货，这是确定不会再更改的，但是对于下周以后的货就算是未来的定单了。用户后来想把未来的定单也加到真实的定单里，以便工厂能准备多一点产品，但是用户又改变主意，不加了。项目组成员，关于加还是不加这个未来定单产生分歧，发生讨论、争辩，其实问题并不复杂，只是白白浪费了时间。

所以在此次项目，我们引进了 Rational 工具。当然 Rational 是个集成了很多功能的工具，我们使用需求管理这块功能，可以将需求分类、统计、关连有关系的需求，能追溯需求变更的历史。有了这个工具，项目组就很少讨论、争辩，大家都能共享这些数据随时取得最新最正确的需求。

##### （2）用户需求变更

用户需求经常变动，是很常见的。究其原因，是用户向我们提供的需求经常是片面的，因为用户没有亲身体验最终系统，他们也无法知道那些具体的需求是他们要的。通常我们会拿需求规格说明书与用户讨论每个细节问题。由于是基于文字，有时效果不好。因此我们做了个原型让用户先期体验，虽然作原型花费了些时间，但是值得的。因为用户通过对原型的了解，就能更进一步的确定自己所需求的最终产品，而不必总是用空洞文字来描述具体产品的模型。还有一些需求变更，是用户自身原因，如他们当初没有考虑清楚，随便跟我们谈了，事后又觉得不妥，要求更改设计。如：用户跟我们讨论如何跟国外工厂接口时，工厂需要这边在固定的时间把数据发过去，要求的是每月第四个工作日，，当初给我们提到是第四个工作日，是按用户自己内部的日历计算的。一直到集成测试时，我们才发现工厂要求的第四个工作日是按照自然日历计算的。由于我们使用了 Rational 需求工具，通过查询，查出原来客户的需求，用户最终承认没有说清楚，使我们能拿到时间去修改新的需求。

## 2. 程序 Bug 的跟踪

从程序开始单元测试直到最后的用户测试，都伴随着对程序 Bug 的查找和修改。但是在项目组里同样的 Bug 重复出现。这中间的问题有可能是没有按照编程规范造成的，也有碰上了技术上的问题。就如 COM 对象的释放问题。在 .Net 里，托管程序是不会去释放外部引用的 COM 类的对象的，需要手工释放，并要手工调用垃圾回收，否则，每个 COM 对象都没有释放，一直占据着内存，直至内存不足至系统崩溃。

更多的问题，是有些程序员碰到过的一些问题，他自己已经解决了，后来这些问题又出现在其他程序员身上，其他的程序员不知道这些问题已经解决了，而去花费了很多时间去找解决方法。特别是到了工程后期，陆陆续续完成工作的成员会撤出，会给继续工作的成员带来不必要的时间上的花费。

为此，为了跟踪 Bug 及各种未完事项，我们引进了 Jtrac 工具，它是个开源的 Bug 跟踪工具，通过它，我们可以在发现 Bug 后，将它们详细记录在 Jtrac 的数据库里，此工具还能通过 email，把 Bug 的相关信息发给相关人。在解决完这些 Bug 后，记录下详细的解决过程，并将关闭这些 Bug 记录，会自动归类到已解决的 Bug 类里，同时也会发 email 给相关人员。在今后发现 Bug 时，可以通过其搜索功能检索是否已经出现过类似问题，并找到其解决方法。此工具能对 Bug 进行各种分类、统计等功能。

虽然 Rational 也有 Bug 跟踪工具，由于 Rational 比较昂贵，而 Jtrac 是开源而且免费的，自然就选用 Jtrac，当然，开源免费的 Bug 工具不止这一个，还有很多。

在基于以上的讨论，在我们改进软件过程中，使用有效的工具管理是很有必要的。Rational 是个很不错的工具，它提供了从需求、分析到设计的功能，还能生成程序的框架。Rational 不足的地方是它比较贵，在小型项目上使用不是很经济的。

以上讨论的关于两个在软件过程中需要改进的地方，只是软件过程改进中的一部分，在整个项目中还会有许多需要改进的，需要我们在实践的基础上，发现它们，并找到解决方法，改进它们。