**如何编写有效用例**

　　用例是什么？用例是代表系统中各个项目相关人员之间就系统的行为所达成的契约。用例描述了在不同条件下，系统对某一项目相关人员的请求所作出的响应。提出请求的项目相关人员被称为主执行者。主执行者通过发起与系统的一次交互来实现某个目标。系统对任一执行者所做出的响应，要保证所有项目相关人员的利益不受侵犯。根据执行者做出的请求和请求涉及的条件，系统将执行不同的行为序列，每一行为序列称之为一个场景。一个用例是不同场景的集合。

　　由于用例是为了让不是精通计算机专业的人看的，所有用例应该尽可能的简单，虽然有时候会用流程图、顺序图等表示用例，但是为了保证没有受过专门培训的人员也能看懂，通常用简单的文本进行编写用例。

　　一个编写良好的用例应该具有很好的可读性。它由多个句子组成，所有句子都采用一种语法形式——一个简单的执行步骤。通过执行这些执行步骤，执行者或者获得一定结果或者向另一个执行者传递信息。可以很简单的学会阅读用例。

　　然而，编写一个好的用例是不容易的。首先要掌握三个概念：范围、主执行者、层次。

　　范围：真正被讨论的系统是什么？

　　主执行者：谁有要实现的目标？

　　层次：目标的层次是高、还是低？

　　执行者要有目标，但是这仅仅解释了如何编写用例中的句子，但是没有涉及如何描述所讨论系统的内部行为方面的内容。所以执行者和目标模型需要基于“用例是具有利益的项目相关人员之间的契约”这个观点而进一步被扩展。扩展后的模型称为“项目相关人员和利益”。

　　实际上，并不需要对每个用例都从头至尾单独的进行描述。这是一个很糟糕的编写策略，因为这样做既乏味又容易导致内容冗余，此外还很难维护。

　　用例，作为规范行为的契约，捕获了与满足项目相关人员利益相关的所有行为，并且仅限于此。要认真的完成一个用例，有必要列出所有的项目先关人员，然后根据用例中的操作命名他们的利益，声明如果用例成功完成将对每一个项目相关人员以为这什么以及靠什么保证他们能从系统中得到他们所希望得到的利益。把这些搞清楚之后，就可以开始编写用例中的执行步骤，确保从用例被触发开始直到其结束期间所有这些不同的利益都能得到满足。如此，就能知道编写用例时从哪里开始到哪里结束，以及用例中包含什么，不包含什么。

　　但是许多人并没有这么认真的编写用例，但是往往也能很幸运的完成任务。好的用例编写者在写随意型用例时，往往是在自己的头脑中做这些练习。他们可能会忽略某些东西，但是被忽略的部分在进行软件开始时会有办法重写捕获得到。这种做法在许多项目中都进行得很顺利，但是有时却会招致巨大的损失。

　　为了满足项目先关人员的利益，需要描述三种行为：

　　两个执行者之间的交互（为了促进一个目标）；

　　确认（为了保护项目相关人员）；

　　每部状态变化（代表项目相关人人员）。

　　功能范围是指系统要提供的服务，它最终应被用例所捕获。然而，在项目刚刚启动时，我们并不能确切地了解项目的功能范围。在识别用例的同时也在决定项目的功能范围——这两个任务是密不可分的。

　　设计范围是系统的区域——如果软件占空间的话，可以说成是“空间区域”。它是开发人员负责设计和讨论的系统的集合，包括硬件和软件系统；它是集合的边界。一般当只要说“范围”时，指的是“设计范围”，这是因为功能范围有执行者-目标列表和用例就可以充分地进行定义，而设计范围则是每个用例中非常关心的。

　　项目相关人员是对用例的行为具有特定利益的人或物。当然，每个主执行者都是一个项目相关人员。用例的主执行者是请求系统提供一项服务的项目相关人员。对系统来说，它有一个目标——系统通过执行操作可以实现该目标。主执行者经常是触发用例的执行者。

　　主执行者在需求收集工作刚开始时和系统将要发布之前一段时间内是重要的。而在这两个时间点之间的那段时间里。它们是相当不重要的。

　　三个命名的目标层次包括用户目标、概要目标、子功能。

　　用户最感兴趣的是用户目标。它是主执行者努力是工作得以完成的目标，或是用户使用系统的目标。它相当于业务过程工程中的“基本业务过程”。

　　概要层次目标包含多个用户目标。

　　子功能层次的目标是指那些在实现用户目标时可能会被用到的目标。只有当迫不得已时才会把它们包含进来。