## [软件需求规格说明书的评审检查单](https://wyw5257.lofter.com/post/1cd587f1_4f36193)

软件需求评审，作为一种软件产品验证的活动之一，通过及早地从软件产品中识别并消除缺陷，从而减少后期的返工，加快开发进度，提高产品的质量。

在需求阶段，发现一个需求缺陷的价值是多大呢？业内有个缺陷修复成本比例，

　　需求阶段：设计阶段：测试阶段：上市阶段＝N:10N:100N:1000N；

方案一

一、 注意对需求规格说明的正确性进行评审

需求规格说明的正确性通常可以从如下方面得以体现：

1 是否有需求与其他需求相互冲突或者重复？

2 是否清晰、简洁、无二义地表达了每个需求?

“清晰”是让人能够读懂；“简洁”是让人愿意去读；“无二义”决定”读”的效果,是让大家对需求描述的理解能够达成一致 。

3 是否每个需求都通过了演示、测试、评审，分析是否得到了验证?

4 是否每个需求都在项目的范围内?

5 是否每个需求都没有内容和语法上的错误?

6 在现有的资源内, 是否能实现所有的需求?

7 每一条特定的错误信息，是否都是唯一的和具有含义的?

二、 注意对需求规格说明的实践性进行评审

所谓实践性是指需求本身是否来源于目前企业的相关业务规则和文件制度,而非源于分析师们经验主义的臆测。实践性是判断需求规格说明是不是理论联系实践、密切和用户联系的一个关键性指标。

三、 注意对需求规格说明的完整性进行评审

我们经常由下面的问题清单来评审需求说明书是否”完整” 。

1 编写的所有需求，其详细程度是否一致和合适?

2 需求是否能为设计提供足够的基础?

3 所有对其他需求的内部引用是否正确?

4 是否包含了每个需求的实现优先级?

5 是否定义了功能说明的内在算法?

6 是否包含了所有已知的客户需求或系统需求?

7 是否遗漏了必要的信息?如果有遗漏的话,把他们标记为待确定的问题(TBD) ?

8 是否对所有预期的错误条件所产生的系统行为都编制了文档?

需求说明的完整性主要体现在需求说明的详细程度上，我们怎样判断该需求的描述是否详细呢?我认为需求需要精化，而不是仅仅提出精化功能、对象要考虑涉众参与者、做些什么、需要什么数据信息、受什么业务规则和条件限制、系统会有什么响应，等等。

四、 注意对需求方案的可行性和成本预算进行评审

五、 注意对需求的质量属性进行评审

我们需要评审需求规格说明是否合理地确定了所有的性能目标，是否合理地确定了安全性方面要考虑到的问题。

六、 注意对需求的可实施性进行评审

是否对每个需求都设置了惟一性并且可以正确地识别它？是否每个功能需求都可以跟踪到高层需求（比如系统需求或用例）?

需求必须可以测试,每个需求在特定的输入条件下应当能给出已知的输出结果。同时,需求应当层次分明,需要把单个需求下面的相关需求综合在一起形成一组需求功能。

需求的可实施性除了可跟踪性还包括可测试性。事实上, 分析人员和测试人员在编写代码以前把需求模型,分析模型和测试用例综合起来通盘考虑，检查出遗漏的、错误的和不必要的需求。软件需求在概念上的测试是一种很必要的技术，它可以在项目早期阶段发现需求的歧义和错误。

七、 注意对需求包含的用例文档进行评审

用例是参与者对系统和参与者的交互过程所达成的一种契约。需求说明书基于用例的分析方法是也是当前较为流行的需求开发方式。用例文档作为需求重要的成果性文档也是需求评审主体之所在。需求评审确认的重点是对关键用户的最常用和最重要的用例进行深入和细致的评审,首先要通过测试用例的主干过程。而我们是否撰写有效的用例则要从以下方面着手评审。

1 用例的目标或价值度量是否明确?

2 用例是否是独立的分散任务?

3 是否明确说明可用用例会给哪些参与者带来用处?

4 编写用例的详细程度是否恰当?是否有不必要的设计和实现细节?

5 所有预期的分支过程是否都编写了文档说明?

6 所有预估的异常过程是否都编写了文档说明?

7 是否存在一些普通的动作序列可以分解成独立的用例?

8 每个路径的步骤是否都清晰明了,无歧义而且完整?

9 用例中的每个参与者和步骤是否都与所执行的任务有关?

10 用例中定义的每个可选路径是否都可行和可验证?

11用例的前置条件和后置条件是否合理?

分析师必须确认用例的前置条件和后置条件准确界定了用例的边界范围,区分了用例和用例之间的界限。

八、 注意需求评审会的过程和结束标准

需求评审会的结果是对需求规格书完成了评审过程,那我们又如何判断审查的结束标准呢?请看如下几条建议:

1 审查期间评审员们提出的所有问题都已经解决。

2 相关文档中的所有更改都已经正确完成。

3 修订过的文档进行了拼写检查。

4 所有标识为TBD(待确定)的问题已经全部解决, 或者已经对每个TBD的问题的解决过程、计划解决的目标日期和责任解决人等编制了文档。

5 需求文档正式进入了配置库。

方案二

组织和完整性

\* 所有需求的编写在细节上是否都一致或者合适？

\* 是否包括了每个需求的实现优先级？

\* 软件需求规格说明中是否包括了所有客户代表或系统的需求？

\* 是否在需求中遗漏了必要的信息？如果有的话，就把它们标记为待确定的问题。

\* 是否记录了所有可能的错误条件所产生的系统行为？

正确性

\* 是否有需求与其它需求相冲突或重复？

\* 是否简明、简洁、无二义性地表达每个需求的？

\* 是否每个需求都能通过测试、演示、审查得以验证或分析？

\* 是否任一个特定的错误信息都具有唯一性和明确的意义？

质量属性

\* 是否合理地确定了性能目标？

\* 是否合理地确定了安全与保密方面的考虑？

\* 在确定了合理的折衷情况下，是否详实地记录了其它相关的质量属性？

可跟踪性

\* 是否每个需求都具有唯一性并且可以正确地识别它？

\* 是否可以根据高层需求（如系统需求或使用实例）跟踪到软件功能需求？

特殊的问题

\* 是否所有的需求都是名副其实的需求而不是设计或实现方案？

\* 是否确定了对时间要求很高的功能并且定义了它们的时间标准？

进入和退出审查的标准

当软件需求文档满足特定的前提条件时，你就可以进行需求审查了。这些标准还可以使审查小组避免把时间浪费在审查之前就应该解决的问题上。调解者在决定进行审查之前，可以把进入审查的标准作为一种清单，并以此作为判断的标准。

\* 文档符合标准模板,并且已经做过拼写检查和语法检查。

\* 在文档中打印了行序号以方便在审查中对特定位置的查阅。

\* 所有未解决的问题都被标记为（待确定）。

\* 包括了文档中使用到的术语词汇表。

相似地，在调解者宣布审查结束之前，你应该定义所满足的退出审查的标准。

\* 已经明确阐述了审查员提出的所有问题。

\* 已经正确修改了文档。

\* 修订过的文档已经进行了拼写检查和语法检查。

\* 所有（待确定）的问题已经全部解决，或者已经记录下每个待确定问题的解决过程，目标日期和提出问题的人。

\* 文档已经登记入项目的配置管理系统。

评审形式一

需求评审可以分为正式评审与非正式评审，在需求规格说明书完成后，需求组必须自己对需求做评审。如果需求组递交的需求规格说明书在指导后面的工作的时候出现很明显的错误，我想拿高工资的需求分析人员是无法向老板交差的。为了需求分析人员的名誉，他们自己会对自己提交的内容进行审核，直到他们认为自己的工作成果足够好，才会将需求规格说明书提交给正式评审组。

    正式评审组的成员一般由公司内经验最丰富，技术最牛的人（技术总监）来担任，当然参加评审的人中间还应该有项目经理、QA人员、测试人员、架构师，他们仔细阅读需求规格说明书，并针对自己将要开展的工作内容进行检查，并提出问题

    正式评审是最后一关，如果正式评审通过了，将进入系统设计阶段，如果在系统设计阶段再跨里程碑来修改需求的话，所花费的代价将大大增加。因此正式评审将是一个“鸡蛋里挑骨头”的过程，只有所有的人都认为需求已经没有什么可挑剔评审才能通过。

评审形式二

目的：为了检查需求的合理性和可实现性。

参与人员：需求评审的参与人员主要包括：

  --需求提出方：产品经理或者第三方

  --系统架构师：架构师可以更好的把握系统架构，检查需求的可实现性

  --开发工程师：开发工程师最了解需求实现难度和是否能实现

  --测试工程师：从测试的角度评价需求合理性

  --客户服务人员：客户和最终用户距离最近，对一般用户了解较多，因此客服人员可以更好的站在用户角度考虑问题

需求评审的准备和进行：需求评审之前，需求提出方最好提前把需求文档发给需要参会的相关人员， 让大家有时间考虑的更深入和更全面

   召开需求评审的会议时，对于一致通过的需求，可以过速的通过，对于各方有疑问和提出疑义的问题，需要一起讨论，给出大致的方案。

   需求评审完成后，需要会议记录人员发送纪要给各方，并发需求的问题和解决方案写清楚。 会议上没有解决的问题，如果是小问题，可以通过邮件方式讨论，如果需要，可以再次召开需求评审会议