这是民兵编组系统项目组的标准调研报告，我们组从**软件能力成熟度模型**、**面向对象的软件系统建模文档编制规范**、**软件测试**、**质量模型和质量保障计划**五个方面结合我们的项目做了一些调研:

**软件能力成熟度模型调研报告**

131220026 杨月洋

我所讨论的标准为电子行业标准中的《软件能力成熟度模型》，代号为“SJT 11235-2001”。

《软件能力成熟度模型》，可简称为CMM。它以模型的方式定义了软件的开发过程，根据过程中所体现的改进能力与成熟度将其分为5个层级，适用于第三方组织对软件开发能力及成熟度进行评估，也利于组织对于自身的开发过程持续改进。除此之外，CMM指定了一些开发中的“关键过程域”，并将其作为是否达到某一CMM级别的衡量。

本小组的课程实验内容为开发一款“民兵管理系统”软件，如何在开发过程中遵从该标准，满足相应的要求，则是我所考虑的内容。本组实验采用传统开发方式，由于开发时间相对宽松故没有选择敏捷开发，但需要对软件开发的每个步骤做到详尽与完备，而这也适合于CMM中的“关键域”的讨论（若采用敏捷开发方式，可能不完全适合于CMM的要求）。

本小组的成员大多没有什么项目的开发经验，因此无法满足过高的标准要求。CMM1级描述的是一个相对混乱和无序的开发过程，强调的是个人能力的发挥，并不提倡进行。在此，我们仅以CMM2级为目标要求，争取满足相应的过程要求。

CMM二级的关键过程域包含**需求管理、项目策划、项目监督和控制、供方协定管理、测量和分析、过程和产品质量保证以及配置管理**。本小组在软件开发过程中，应尽量实现每一个过程，满足相应的要求。

**需求管理**可以说是一切其他过程的基础，保证需求被正确理解并被成功实现，是软件质量的保障，同时对于需求的变更也应有相应的准备。对于本小组，开发成员应与客户（本项目为指导教师）进行沟通，对软件项目的产生相同一致的理解，对软件开发的每一步都尽量得到客户的肯定，甚至可以将客户拉入软件的管理团队之中，以便于更好的控制整个过程的进行。

**项目策划**为软件开发制定合理的计划。对于本次开发过程，本小组应制定合理的开发纲要，明确时间规划，即在什么时间做什么事情。对开发的每一个步骤如需求分析、概要设计、详细设计、测试及配置等内容都有详尽的规划。

**项目监督和控制**可以防止软件的实际开发过程与预期情况相违背。对于本小组中，应设定直接的项目监督人，暂指定为组长、助教和指导教师。定期将软件的开发进展对其进行汇报，并及时得到反馈，对于与预期情况不符之处做出处理和有效控制。

**供方协定管理**负责的是软件的采购和外包等内容。在本小组的开发过程中并不涉及这部分内容，故暂时不作考虑。

**测量和分析**对软件开发每一步骤的临时结果进行评估，对于软件开发的工具、方式及方法进行相应的比较和分析，选择最为合适的方案。当遇到分歧或面临多个选择时，尽量选择最优方案。在实际开发过程中，我们小组会定期举行讨论会议，以对每个问题得到处理结果，每个小组成员发表个人意见，最终由组长定夺，并将组长作为最终责任人，每个人作为自己负责模块的责任人。

**过程和产品质量保证**，这不仅需要完善的审查与监督，更需要完备的文档化的过程描述。对于每一个步骤，都应有相应的文档化的规约。如需求分析阶段应有需求文档，设计阶段有设计文档，测试阶段应有测试文档等。对于不同软件模块的接口，应有相应的文档进行描述和限制。在我们的开发过程中，需要用到非关系型的数据库，XML格式的数据文件，便于使用的windows平台的用户界面，对于这些模块如何实现与交互则需要我们以一种精确的方式描述，最好的方式便是以文档方式进行规约和限定。

**配置管理**的目的是在软件生命周期中建立和维护产品的完整性。高效的配置管理能够使软件因需求、方法变更而造成的问题最小化，即合理的控制各种情况下的开发成果。在我们的软件开发过程中，首先对于需求的限制应明确规范，当出现任何变更时都要考虑到整体的影响。提交后的软件成品都要有版本记录，便于进行版本控制。同时应建立一套系统、全面的软件测试用例，便于对每次的修正进行检验。对于配置过程、工作空间，应有专门的人士负责，其他人在每次修改前都应上报并得到批准。我们小组将软件的代码、测试及文档选择放在github上，从而便于进行版本控制和管理。

CMM二级的要求在工业界不算较高的标准，但对于我们开发小组也不是容易实现的。我们需要对于每一个环节都精益求精，并严格按照相应的要求执行，才有可能最终满足其中的标准。

对于如何评价我们的软件开发过程是否完成软件相关的标准，即软件成熟度模型的要求、软件过程域的要求，我们将完全按照“SJT 11235-2001”的描述进行比对，这也是我们的唯一参考“蓝图”。当然，由于我们团队并不是正规的开发团队，对于一些具体的要求我们可能无法完全实现，这则需要斟酌处理，争取满足开发过程中应该实现的那些要求的“关键过程域”。

“哪些软件工程活动或结果缺乏标准支持”这个问题对于我这个没什么实践经验的人来说不太了解。但在我看来，传统的软件开发过程的标准有不少，并且对每一个阶段、步骤或者说“过程域”都有相应的描述，而在敏捷开发行业，可能相关的标准并不算多。

敏捷开发注重效率，使开发过程变得简单实用，便对于一些内容要求得并不严格。可能相关的“原则”不少，但实际的有详细描述的“标准”并不多。当然，这不是说敏捷开发不注重标准，比如在某些地方，敏捷开发不要求编写很多的文档，当更加强调某些文档的编写规范。我想，对于“敏捷开发”甚至是“极限编程”，或许可以多制定一些完善的标准，既不会影响到开发效率，又可以在业界中建立详尽的规范约束，便于软件产品的规范化。

软件工程标准报告

——面向对象的软件系统建模规范 第3部分 文档编制（SJ/T 11291-2003）

By 131220065 何志强

1. **概述**

这一系列中，第1部分的概念和表示法规定了图形文档的绘制方法，大略看了下，定义的概念很多，涵盖了所有的UML图相关的概念，如果要用一个图或者看懂一个纯粹的UML图可以去查询它的用法。然而正如肢体语言与口头语言的关系，只用口头语言也难以描述得特别形象清楚，加上肢体语言就更利于理解了。同样，纯粹的UML图也是难以全部描述清楚的，尤其是图中的概念如果不熟悉的话还需查询，所以加上文字描述，会更利于人理解一些。而这个文档编制部分的建模规范，就主要说了文字部分的编制规范，目的就是为了让阅读者更容易理解图形文档的内容。

这个标准要求用图形文档和文本文档两种表现形式，且力求图形文档中的元素和文本文档中的各组成部分一一对应。在每种文本文档中对各个组成部分分类描述，并注重描述不同组成部分的关系，这样就使得文本文档中的组成部分组织严密，便于相关人员从不同的角度使用所需要的组成部分。

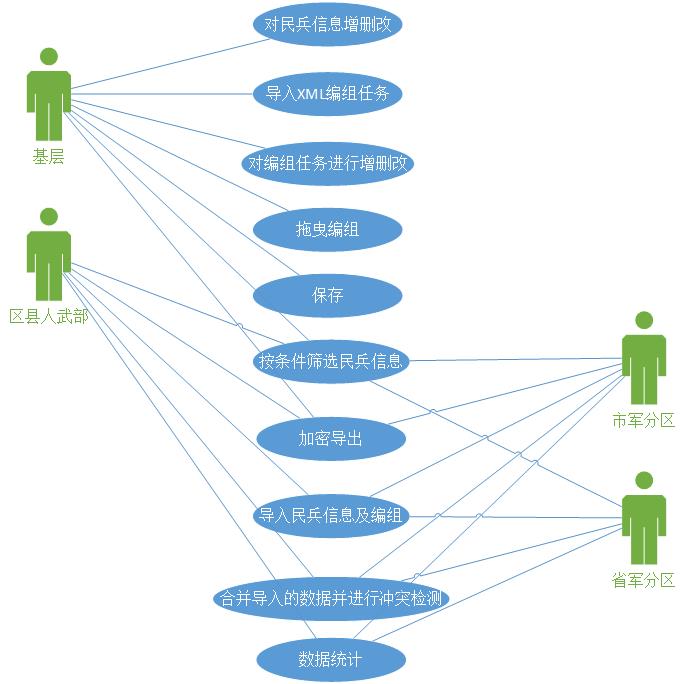
至于为什么文本文档的编制需要有这个标准，我想上面说了几个原因。一个是图形文档中元素与文本文档中元素一一对应，这样保证了文本文档的一致性，这是需要保证的；一个是文本文档中的组成部分组织严密，便于相关人员从不同角度使用所需要的组成部分，即是有了清晰、固定的结构，便于查找与复用。如果没有这个标准或约定，那么可能图形文档难以看懂，如果不对应就会显得前后矛盾，而没有结构则会很混乱，让人感到厌烦。

开发者按照《面向对象的软件系统建模规范 第2部分：过程指导》开发软件系统时，可以按开发阶段组织文档，分别形成：需求定义文档、分析文档和设计文档。其中每种文档的内容包含一个或多个下面的各种图的文档：用况图、类图、顺序图、协作图、状态图、活动图、构件图、部署图。其中有的文档有时也会引入包图来管理信息组织的复杂性。于是此标准就规定了这些图文档的结构以及可能包含的组成部分。

1. **实例**

比如用况图（我想就是用例图）文档，需要有：用况图综述（阐述图的目的、结构、功能等）、参与者描述（附有必要的文字说明）、用况描述（记录用况名称、行为描述、用况图中元素关系描述、其他说明）这些信息。

就我们的民兵编组系统来讲，正处于第一个阶段——需求分析阶段。我们做了用例图（如下），确实没有符合标准的文本文档说明，但从大体上还是能看出这个简单的用例关系的，只是缺少了用况的细节说明。下面我将试着按照本文的标准来完成这个用况图文档：



用况图综述：本图目的是为了从以用户和功能的角度来描述民兵编组系统的需求，结构上分为用户和功能两部分，用户与功能相连表示用户具有这样的功能。

参与者描述：

基层：最底层的用户，拥有最多的功能

区县人武部：管理基层

市军分区：管理区县人武部

省军分区：管理市军分区

用况综述：

对民兵信息增删改：增删民兵信息，修改民兵的所有属性

导入XML编组任务：导入XML格式的编组任务

对编组任务进行增删改：因需求变更，待定

拖曳编组：通过拖曳的方式进行编组

保存：退出后再打开保证和退出前的内容一样（已编组的和未编组的民兵信息）

按条件筛选民兵信息：按照民兵属性，如筛选年龄在28-35岁之间的党员等

加密导出：加密导出民兵编组信息及其编组

导入民兵信息及编组：解密导入上一级导出的民兵信息及它们的分组

合并导入的数据并进行冲突检测：按照身份证号进行冲突检测

数据统计：统计各个属性的民兵的人数，如xx市的民兵人数

1. **总结**

经过实例的实践，发现其实我们的项目的一些用况难以描述清楚，说明我们的需求分析在一些功能上并没有形成一致、清晰的结论。且我们项目的一些图确实比较简单，用况并未涉及元素及元素间关系，或者说有些东西并没有提取出来。可见要实现某个标准也是需要一番分析和尝试的。等到设计阶段也许那些用况会有清晰化的强烈需求，我们现阶段无法标准化，但我们应朝着标准靠拢，也许会更好一点。

**软件工程标准调研报告**

**——计算机软件测试规范**

王凯军 121130086

**前 言**

测试是计算机软件开发过程中的重要环节，它不仅能验证软件是否满足开发合同或者项目开发计划、系统/子系统设计文档、软件需求规格说明、软件设计说明和产品说明等规定的软件质量要求，还可以通过测试发现软件缺陷，以及为软件产品的质量评价提供依据。软件测试始终伴随于开发全过程，有效的测试可以提高开发效率、减少产品漏洞、完善功能需求。因此，我选择了GB/T 15532-2008 计算机软件测试规范作为本次调研报告的主要内容。

GB/T 15532-2008 计算机软件测试规范（以下简称“规范”）规定了计算机软件生存周期内各类软件产品的基本测试方法、过程和准则。这一标准在时间上适用于计算机软件生存周期全过程，在对象上适用于计算机软件的开发机构、测试机构及相关人员。我们小组开发的“民兵编队系统”软件同样适用这一规范。

**测 试 流 程**

每一次软件测试过程都应该按照一定的流程展开，一次完整的测试过程一般包括四项活动，即测试策划、测试设计、测试执行和测试总结。

测试策划作为测试过程中开始的一步，主要是进行测试需求分析，即确定需要测试的内容或质量特性，确定测试的充分性要求，提出测试的基本方法，确定测试的资源和技术需求，进行风险分析和评估，制定测试计划（含资源计划和进度计划）。对于我们来说，每一次测试进行之前必须考虑清楚本次测试的原因和目的，杜绝无意义、无准备测试，力求每一次测试都能起到应有的作用，从而有效地提高效率。

在测试设计阶段，需要依据测试需求，分析并选用已有的测试用例或者设计新的测试用例，获取并验证数据，根据测试资源、风险等约束条件确定测试用例执行顺序，获取测试资源，开发测试软件，建立并校准测试环境，进行测试就绪评审，主要评审测试计划的合理性和测试用例的正确性、有效性和覆盖充分性，评审测试组织、环境和设备工具是否齐备并符合要求。测试的设计必须充分周密，这一角度上讲测试本质上就是另一次开发过程，只不过和开发的思路不同的是，测试是结果驱动而非过程驱动的。从实际情况出发，我们民兵编队系统的开发团队人员相对较少，一般不会设置专门的测试人员，所以在每一次测试活动里都需要尽量多的参与率，覆盖尽可能多的用例和数据。

之后是测试执行阶段，需要执行测试用例，获取测试结果，之后进行分析与判定。同时，根据不同的判定的结果采取相应的措施，对测试过程的的正常或者异常终止情况进行核对，并根据核对结果，对未达到测试终止条件的测试用例，决定是停止测试，还是需要修改或者补充测试用例集并进一步测试。测试执行阶段对每一个测试人员要求最高的是专注度和细致心，认真对待每一个测试样例和数据，不漏过任何一个数据和结果，对于随时可能发生的现象和变化保持高度的专注。在本次课程项目——民兵编队系统中，无论是数据库的组织、前台功能的搭建还是UI的设计，无不需要根据以上诸项要求进行测试。

最后是测试总结阶段，主要是整理分析测试数据，评价测试结果和被测软件项，描述测试状态。测试总结阶段是一个测试的收尾工作，在这个阶段我们需要做的是整理测试过程中出现的数据、现象和结果，并加以分析。囿于开发时间周期所限，我们很难做到使用特制的测试程序进行对软件的测试，绝大多数的测试输入输出和结果分析只能人工进行，这就需要我们对测试结果和数据的把握能力更强，对测试过程和软件开发需求的理解更为深刻。

**具体的测试形式**

根据方法、对象和阶段的不同，测试被分为单元测试、集成测试、配置项测试（也称软件合格性测试或确认测试）、系统测试和验收测试。除此之外，还有可出现在上述每个类别中并贯穿于整个软件生存周期的回归测试。以上测试可根据软件的规模、类型、完整性级别选择执行不同的测试类别。民兵编队系统属于模块相对固定、逻辑相对明确、算法相对简单的软件，针对这一特点，我们小组在开发过程中也要结合实际情况，合理选取相应的测试手段，以期达到降低缺陷度、提高易用性和完善各方面功能的结果。

单元测试的对象是可独立编译或汇编的程序模块，目的是检查每个软件单元能否正确的实现设计说明中的功能、性能、接口以及其他设计约束等要求，发现单元内可能存在的各种差错。在这一阶段的测试中，一般会进行静态和动态两种类型的测试，其使用方式随具体代码内容而异。对于这一测试阶段的主体，规范中规定一般由软件的供方或开发方组织并实施单元测试，对于作为开发者的我们来说，有必要在这一层次上首先进行测试，保证在程序的基础模块中能够实现所需的功能，不出现差错。

集成测试包括对任意一个软件单元集成到计算机软件系统的组装过程的测试和对任意一个组装得到的软件系统的测试，其目的是验证软件单元之间，软件单元和已集成的软件系统之间的接口关系，并验证已集成的软件系统是否符合设计要求。这一阶段的测试相对上一阶段而言测试客体更为复杂，因此也需要不一样的执行方式。这一阶段的测试仍然是由软件供方组织，测试人员与开发人员应相对独立。对于我们来说，虽然由于人数限制难以设置专门的测试人员，在分工中不负责该部分开发的同学都应该参与到这一阶段的测试。

在配置项测试这一环节中，测试的是为独立的配置管理而设计的并且能满足最终用户功能的一组软件，也称软件配置项。这一阶段，软件已经稍具雏形，目的则改变为检验软件配置项和软件需求规格说明的一致性。这一阶段的测试人员同样是软件供方，同样需要软件开发人员进行配合。这一阶段的测试相对前两个阶段而言，重点从对细节组成转向整体功能，如易理解性、易操作性等面向使用者的特性也要作为测试的内容。这一阶段，需要我们从使用者的角度出发，以需求为突破口，实际应用环境为准绳，统一布置开展测试工作，在软件可以正常运行的前提下提高易用性和用户友好度。

系统测试和验收测试的测试对象都是完整的计算机系统，由于我们开发的软件规模较小，尚不能作为系统级的程序，这两部分对于我们来说接触的机会相对比较少。作为测试的组成部分，其中还是有一些流程和思想值得我们学习。

回归测试并不是一个独立的测试阶段，而是贯穿于各个测试阶段中的补充性内容。未通过各阶段测试的程序、单元和代码，在变更之后需要进行回归测试。回归测试的具体流程因测试对象而异，不同阶段的回归测试参考前述流程，这里不再赘述。

**总 结**

测试作为软件开发过程中的重要活动，贯穿于整个开发进程中，无论哪个阶段都需要测试的进行。测试有助于降低软件漏洞、优化实现方式、完善产品功能。对于我们——民兵编队系统程序开发团队来说，一定要结合现实特点，选择合适的测试手段和流程，在测试时保持高度的专注和细致，力争每个测试都能达到需要的目的，为整个软件的开发打下坚实的基础。

# 软件工程产品质量 第1部分：质量模型阅读报告

131220064 刘闯

* GitHub账号：ChuangLiu27（[15850538707@163.com](mailto:%31%35%38%35%30%35%33%38%37%30%37@%31%36%33.%63%6f%6d)）

## 阅读内容总结

这次阅读的软件工程标准是“GB-T 16260.1-2006 软件工程 产品质量 第1部分：质量模型”。

软件产品质量可以通过测量内部属性（典型地是对中间产品的静态测度），也可以是通过测量外部属性（典型地是通过测量代码执行时的行为）。一般意义上来说，广义的软件产品质量应该是有递进关系的三部分所组成的，它们是过程质量，软件产品质量，使用质量。这个划分是符合人的一般认知的，前者有助于提高后者水平，而后者也可以为改进前者过程提供反馈。

标准中描述了关于软件产品质量的两部分模型：a）内部质量和外部质量，b）使用质量。

内部质量是基于内部视角的软件产品特性的总体，外部质量是基于外部视角的软件产品特性的总体。使用质量是基于用户观点的软件产品用于指定的环境和使用环境时的质量。

**外部和内部质量的质量模型可以划分为六个特性及细分后的子特性。它们是：**

**功能性：**当软件在指定条件下使用时，软件产品提供满足明确和隐含要求的功能的能力

①适当性：软件产品为指定的任务和用户目标提供一组合适的功能的能力

②准确性：软件产品提供具有所需精度的正确或相符的结果或效果的能力

③互操作性：软件产品与一个或更多的规定系统进行交互的能力。

④软件保密性：软件产品保护信息和数据的能力，以使未授权的人员或系统不能阅读或修改这些信息和数据，而不拒绝授权人员或系统对它们的访问

⑤功能性的依从性：软件产品遵循与功能性相关的标准、约定或法规以及类似规定的能力

**可靠性：**在指定条件下使用时，软件产品维持规定的性能级别的能力

①成熟性：软件产品为避免由软件中故障而导致失效的能力

②容错性：在软件出现故障或者违反其指定接口的情况下，软件产品维持规定的性能级别的能力

③易恢复性：在失效发生的情况下，软件产品重建规定的性能级别并恢复受直接影响的数据的能力

④可靠性的依从性：软件产品遵循与可靠性相关的标准、约定或法规的能力

**易用性：**在指定条件下使用时，软件产品被理解、学习、使用和吸引用户的能力

①软件产品使用户能理解软件是否合适以及能将软件用于特定的任务和使用条件的能力

②易学性：软件产品使用户能学习其应用的能力

③易操作性：软件产品使用户能操作和控制它的能力

④吸引性：软件产品吸引用户的能力

⑤易用性的依从性：软件产品遵循和易用性相关的标准、约定、风格指南或法规的能力

**效率：**在规定条件下，相对于所用资源的数量，软件产品可提供适当性能的能力

①时间特性：在规定条件下，软件产品执行其功能时，提供适当的响应和处理时间及其吞吐率的能力

②资源利用性：在规定条件下，软件产品执行其功能时，使用合适数量和类别的资源的能力

③效率依从性：软件产品遵循与效率相关的标准或约定的能力

**维护性：**软件产品可被修改的能力。修改可能包括纠正、改进或软件对环境、需求和功能规格说明变化的适应

①易分析性：软件产品诊断软件中的缺陷或失效原因或识别待修改部分的能力

②易改变性：软件产品使指定的修改乐意被实现的能力

③稳定性：软件产品避免由于软件修改而造成意外活动的能力

④易测试性：软件产品使已修改软件能被确认的能力

⑤维护性的依从性：软件产品遵循与维护性相关的标准或约定的能力

可移植性：软件产品从一种环境迁移到另外一种环境的能力

①适应性：软件产品毋需采用额外的活动或手段就可适应不同指定环境的能力

②易安装性：软件产品在指定环境中被安装的能力

③共存性：软件产品在公共环境中同与其分享公共资源的其他独立软件共存的能力

④易替换性：软件产品在同样的环境下，替代另一个相同用途的指定软件产品的能力

⑤可移植性的依从性：软件产品遵循与可移植性相关的标准或约定的能力

**使用质量的的属性分类为四个特性：**

有效性：软件产品在指定的使用周境下，使用户能达到与准确性和完备性相关的规定目标的能力

生产率：软件产品在指定的使用周境下，使用户为达到有效性而消耗适当数量的资源的能力

安全性：软件产品在指定的使用周境下，使用户为达到有效性而消耗适当数量的资源的能力

满意度：软件产品在指定的使用周境下，使用户满意的能力

## 阅读之后结合本组目前状况的分析

目前本组正在做需求分析和开发工具的学习。同步工具是github，开发语言是C++。需要注意到的质量问题：

（1）：应使得需求分析达到准确性和适当性。需求管理的准确性关乎到软件产品能否提供具有所需精度的正确或相符的结果或效果的能力，目前的需求已经改了两次，不知道会不会有第三次，对需求的深度挖掘也在进一步加深。根据已经从助教手里拿到的parameters.xml和militia.xml以及字段，可以很肯定知道编队系统的显性要求。但是，用户所描述的要求并不总是反映实际客户的需要，因为：（1）用户并不是经常可以意识到自己的实际需要，（2）要求在被说明之后还可能会继续变化，（3）不同用户可能具有不同的操作环境，（4）针对现在的软件，咨询所有可能类型的用户显然是做不到。

基于目前我们只能从助教和老师那里得到具体的项目要求，更深层次的实现需求还需要我们自己去分析。因此，质量需求不可能在开始设计之前就得到完整的定义。然而，尽可能详细地理解实际用户的需要并在需求中表示它们是必须的。就比如民兵编队用户虽然没有提出要实现批量处理的功能但是现实中为了处理多达50万的数据项，批量处理显然是必须的。

（2）：在本项目中使用质量的度量考虑。结合使用质量的定义，对最终用户来说使用质量主要是功能性、可靠性、易用性和效率性。在此基础上，最终用户会使用直观的反映来度量我们做的产品，例如功能要求中的“最终省级合并所有数据的冲突检测时间不大于5分钟”，对最终用户来说4分45秒和5分钟在具体使用过程中的差异不会明显，但是5分15秒和5分钟会有较明显的差别。

对软件维护人员来说使用质量是维护性的结果。维护人员在维护过程中对文档、代码、使用说明等进行再阅读，使用维护性的度量来评价软件产品。

作为项目的开发人员，我们可以根据使用任一质量特性的内部测度的准则值（代码行数、复杂度测试、在走查和Fog索引中发现的故障数）来评价软件产品。

## 阅读中值得思考的地方

在一个软件产品的生存周期内，内部质量、外部质量和使用质量的观点是时刻变化的。在开发的最早阶段，仅能测量资源和过程。当中间产品（规格说明、源代码等）可用时，就可以在选择的内部度量层次上对它们进行评价。

作为程序员，绝大部分包括我，注意到的产品质量还仅仅停留在使用质量这个层次上，思维上觉得只要编程实现出来的功能足以交付使用就足够了，往往这样的思维会使程序的内部质量（可靠性、易用性、效率等）被掩盖在满足功能的急切心理之下。

事实上作为一个大的项目，代码仅仅只是内部质量的一部分。

**计算机软件质量保证计划规范调研报告**

131130022 石佳煊

**一．概述**

软件质量保证（SQA-Software Quality Assurance）是建立一套有计划，有系统的方法，来向管理层保证拟定出的标准、步骤、实践和方法能够正确地被所有项目所采用。软件质量保证的目的是使软件过程对于管理人员来说是可见的。它通过对软件产品和活动进行评审和审计来验证软件是合乎标准的。软件质量保证组在项目开始时就一起参与建立计划、标准和过程。这些将使软件项目满足机构方针的要求。

GB／T 12504一—90计算机软件质量保证计划规范规定了存制订软件质量保证计划时应该遵循的统一的基本要求，适用于软件中特别是重要软件的质量保证计划的制订工作。对于非重要软件或已经开发好的软件，可以采用本规范规定的要求的子集。

**二．主要内容**

验证verification：验证是指确定软件什发周期中的—个给定阶段的产品是否达到在上一阶段确立的需求的过程。

确认validation：确认是指在软件开发过程结束时对软件进行评价以确定它是否和软件需求相一致的过程。

测试testing：测试是指通过执行程序来有意识地发现程序中的设计错误和编码错误的过程。测试是验证和确认的手段之一。

计算机软件质量保证计划包括计划的目的、定义及缩写、参考资料、管理机构和职责、软件配置管理、质量管理工具、计算机程序物理媒体的控制、对供货单位的控制、记录的收集维护保存、评审和检查等，重点是需要下列文档：

软件需求规格说明书software requirements specification必须准确描述软件的每个基本需求和外部界面，必须把每个需求规定成能客观验证的形式。

软件设计说明书software design description必须描述所设计软件的总体结构、外部接口、各主要部件的功能与数据结构、各主要部件之间的接口；并给出每一个基本部件的功能、算法和过程描述。

软件验证与确认计划software verification and validation plan必须描述所采用的软件验证和确认方法，以验证软件需求规格说明书中的需求是否已由软件设计说明书描述的设计实现、软件设计说明书表达的设计是否已由编码实现、编码的执行是否与软件需求规格说明书中所规定的需求相—致。

软件验证和确认报告software verification and validation report必须描述软件验证与确认计划的执行结果。 这里必须包括软件质量保证计划所需要的所有评审、检查和测试的结果。

用户文档user documcntation必须指明成功运行该软件所需要的数据、控制命令以及运行条件等；必须指明所有的出错信息、含义及其修改方法；还必须描述将用户发现的错误或问题通知项目承办单位(或软件开发单位)或项目委托单位的方法。

除基本文档外，还应有项目实施计划、项目进展报表、项目开发各阶段的评审报表和项目开发总结等

1. **阅读之后的总结**

首先，之前我只重视测试却缺乏验证和确认的概念，甚至在以前的大作业中最后的程序偏离了自己开始的计划，勉强符合作业要求就交上去了，这点以后要注意。对多人合作完成的民兵编组系统，一定要在每个阶段后进行检查，防止编码偏离需求。

其次，虽然我们的项目不属于重要软件，可以适当简化，文档仍然数量众多，写代码只是软件工程的一部分，甚至可以说不是最关键的部分。

其三，软件要进行多种评审，例如需求中的“时间不超过5分钟”等需要事先主义。