软件工程标准报告

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。