

第15章

1.什么时候建立术语表？

答：术语表是对重要术语清晰、一致的说明，用于准确描述术语的含义。计划和编写文档时通常会使用一些术语，编写过程中，如果对这些术语要作一定的修改或者可能有心的想法，此时建立并维护一个术语表可以让文档中的术语保持一致，使得文档更容易阅读和维护。

同时，如果用户和文档编写者使用不同的方言，那么可能相同的词汇会产生不同的理解，术语表可以解决方言问题。

如果文档写作中使用了过多的复杂词汇和表达方式，则会提高文档的模糊性，此时使用术语表可以减少冗余，避免术语的错误。

2.在需求获取和需求分析中采用哪些手段可以保证最终需求集的完备性、一致性和正确性？

答：

完备性：需求规格说明文档时完备的，当且仅当

- 1描述了用户所有有意义的需求，包括功能、性能、约束、质量属性和对外接口；
- 2定义了软件对所有情况的所有实际输入（无论有效输入还是无效输入）的响应；
- 3为文档中的所有插图、图、表和术语、度量单位的定义提供了完整的引用和标记。

需求的完备性要求不能遗漏任何需求或者必要的信息，为避免需求遗漏，需求工程师要做好业务需求的分析，建立并控制正确的项目规范，建立业务需求、用户需求和系统需求的跟踪关系也用于发现需求的遗漏现象。文档内所有TBD（待解决问题）被全部解决之前，需求规格说明文档都是不完备的。

手段：

需求获取阶段：防止遗漏需求

- 1.让所有涉众都表达自己的意见
- 2.不要以抽象和模糊的需求作为结束
- 3.使用多种方法表达需求信息
- 4.主义检查边界值和布尔逻辑

需求工程师做好业务需求的分析工作，建立并控制正确的项目范围。

需求分析阶段：

- 1.选取并建立合适的分析模型
- 2.不遗漏非功能性需求
- 3.建立业务需求、用户需求和系统需求的跟踪关系
- 4.将不能定论的内容显著地标记为待解决问题，并指定解决的时间和人员。

一致性：

- 1细节的需求不能同高层次的需求相冲突
- 2同一层次的不同需求之间也不能互相冲突。

也就是说，软件需求规格说明文档既要在所含内容上保持一致，也要和更高层次的文档（例如系统需求规格说明书）所含的内容保持一致。为保证需求规格说明文档的一致性，由开发人员和非开发人员对于其进行手工评审时非常必要的。

在需求获取阶段： 不要以抽象和模糊的需求作为结束

平衡不同涉众之间的共赢关系

主义需求协商，在需求出现冲突的情况下主义明确冲突的因素、明确冲突的解决空间，确定最佳的解决方案

正确性：保证文档中每个单一需求都是优秀的需求。单一需求的优秀特性可以使整份文档满足正确性。

在需求获取和分析阶段，都要做好文档的审查和验证工作。

在需求获取阶段，要做好充分的准备，一方面要明确问题，保证业务需求、项目范围的正确性，一方面要做好涉众分析，从不同层次的涉众中获取需求，确保需求的正确性。

3.关于文档化的三种手段——非形式化、半形式化和形式化，一直以来存在较多的争论，对此你是怎么看的？（08年考题）

答： 形式化语言是基于数学的语言，其描述内容是可以进行逻辑一致性推导和证明的，能够保证信息的正确性。然而不易被用户理解，且描述的内容相对有限。

相反，非形式化语言即自然语言，表达能力最为强大，也最容易被理解，然而它具有松散、模糊、歧义、凌乱等不好的特点，也无法被机器所理解。

半形式化语言基于两者之间。比自然语言具有更丰富的语义和更严格的语法，又没有严格到完全基于数学方法的语言。可以避免自然语言模糊、松散、歧义、凌乱等不好的特性，又可以做到具有丰富的语言，使用的规则比单纯的形式化语言复杂很多，使得其有更强的表达能力。当然，由于它并没有严格到可以等价数学方法的程度，所以必须经过额外处理才能被机器所理解。又因为它是一门全新的语言，具有独特的语法和语义，所描述的信息也难被用户理解。

我的考虑是综合使用相对比较合适，作为用户，显然更希望文档使用自然语言来描述易于自己理解，而作为开发人员，更希望使用半形式化、形式化的语言，更加严格和准确。实际文档编写过程中，可以考虑使用半形式化或形式化语言配上自然语言的注释，来描述某个信息。也可以使用半形式化、形式化语言来描述概要和抽象信息，然后再用自然语言来进行详细信息的描述

第16章

1.用于需求获取的原型和用于需求验证的原型有何异同？（09年考题）

答：都是原型的一种方法，成本都相对较高

用于获取的原型起始于不明确的需求，侧重于可调整方案

用于验证的原型起始于比较明确的需求，侧重于功能评价

2.多种需求验证的方法应该如何结合运用？

答：需求验证的方法包括：需求评审（静态分析，需求验证的一种主要方法），原型与模拟，开发测试用例，用户手册编制，利用跟踪关系，自动化分析

1.首先，每个需求都需要经过评审，对于一些动态行为评审不能完整，就要通过原型和模拟来验证。

2.在正常的工作中，可以顺使用上用户手册、测试用例、跟踪等方法在一些错误之处或者是一些需求上进行验证，也是比较有效的。

3.总之：几乎所有的需求都需要经过评审，个别动态复杂的需求需要使用原型与模拟，工作之间产生的衔接可以用上开发测试用例、用户手册等方法，实现高效的综合运用。

第17章

1.如何有效的处理需求的变化是很多现代软件开发技术的主题，对此现象你有什么看法？结合本章内容，你将怎样做以控制一个需求多变的项目？

书：P392

答： 1.首先，需求的变更是正当和不可避免的。在需求建立的过程中，一些现象例如：问题的改变、环境的变化、需求基线本身存在缺陷、用户的变动、用户对软件认识的变动、相关产品的出现都会导致需求的变更。而针对合理必要的需求变更，必须建立起一套成熟的高效的应对策略，所以对于这个现象，我是认同和理解的。

2.首先需要明确的是，需求的变更必然会给项目带来一定的负面影响，所以随意的变更是不恰当的做法，正确的做法应该是在形成需求基线之后，进行需求的变更控制。其过程包括：

- 1.提请需求变更
- 2.接受变更请求
- 3.变更评估
- 4.变更决策
- 5.执行变更
- 6.验证变更

在变更控制中，需要注意一些事项，包括

- 1.认识到变更的必要性，并为之制定计划。
- 2.维护需求基线，审计变更记录
- 3.管理范围蔓延

对合理的不可不免的变化要进行有效变更，对不合理的请求要敢于说“不”

- 4.灵活应对变更请求

如果需求变更的请求对项目影响重大，但是对用户也相当重要，此时可以重新协商原想的项目约定。

- 5.使用辅助工具。