# 15-17思考题

## 第15章：

1. **什么时候建立术语表？**

项目初期有，需求整个阶段不停维护，

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。通过术语表，文档的写作者和读者可以就术语所表示的意义达成共同理解。定义术语表时可以避免下列常见问题：

术语不一致：在人们计划和编写文档时，通常会使用一些术语，然后无数次地修改这些术语，特别是当他们对属于有新的想法，以及对问题域的理解有更深认识的时候，这种做法带来的负面影响是，在同一个文档中累计了针对同一个事物的不同的、旧的和新的术语定义，这会给读者的正确理解带来困难。如果能够建立并维护一个术语表，那么人们可以很容易的再文档中进行术语的查询和替换，这样就可以避免文档中出现的术语不一致。

方言问题：建立术语表时可以逐一解释术语的“方言”形式，这样就能妥善解决方言问题。

建立属于问题域的术语表可以避免术语问题，在术语表的基础上展开文档的内容描述可以避免冗余术语问题。

**2.在需求获取和需求分析当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？**

解答：

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。

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。

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，无歧义和可验证。正确性是文档内的所有需求都有正确性，无歧义指的问胆囊管选中的指令，可验证文档内所有的所有需求都是可验证的

3. **进行需求文档写作时主要有三种手段：形式化语言、非形式化语言和半形式化语言，请：（1）说明这三种手段的不同及其各自的优缺点，并试举例；（2）说明在实践当中，应该怎样综合运用这三种手段。(考过)**

答案：

信息的描述语言可以分为三种类别：

* 非形式化语言，即自然语言。
* 半形式化语言，比自然语言具有更丰富的语义和更严格的语法同时又没有严格到可以完全基于数学方法的语言，例如ER图、DFD图、UML等图形语言。
* 形式化语言，基于数学的语言，例如VDM、Z语言等。

自然语言具有复杂的规则和多样化的表达方式，所以它的表达能力最为强大。而且自然语言是属于普通人的语言，每个人都熟知其规则、表达方式和特点，所以非常利于用户的理解。但同时自然语言也具有松散、模糊、歧义、凌乱等不好的特性。这使得它无法被机器所理解，它所描述的信息内容也无法准确的映射为机器行为。

形式化语言是基于数学方法的语言，具有数学的表示法特性。使用形式化语言描述的信息内容是可以进行逻辑一致性推导和证明的，所以它能够保证信息的正确性。而且形式化的信息描述能够被机器所理解，它所描述的信息内容可以准确的映射为机器行为。但是形式化描述的信息要求读者具备谓词演算方面的知识，这对普通的用户而言显然要求过高，以至于大多数用户无法读懂以形式化方法描述的信息。形式化方法所能描述的内容也是有限的，具体的有限性因形式化方法的不同而各异。

半形式化语言是介于自然语言和形式化语言之间的描述语言。一方面，半形式化语言具有严格的语法，定义方式比自然语言更加严格，这使得它可以避免自然语言模糊、松散、歧义、凌乱等不好的特性。另一方面，半形式化语言具有丰富的语义，使用规则比形式化语言更复杂和多样，这使得它具有比形式化方法更强的表达能力。但是，丰富的语义使得半形式化语言的语法无法严格到可以等价于数学方法的程度，所以它描述的信息还需要进行额外的处理才能够被机器所理解或者准确的映射为机器行为。同时，严格的语法限制也使得半形式语言的表达能力无法达到自然语言的程度。而且因为具有独特的语法和语义，所以半形式语言对普通用户而言无异于一门全新的语言，它所描述的信息很难被用户所理解。

为了让需求规格说明文档的内容能够同时满足用户和开发人员的需要，需求工程师在实践中更多的会综合使用自然语言、半形式化语言和形式化语言。例如，为半形式化语言和形式化语言添加自然语言的注释，或者分别使用自然语言和半形式化语言（或者形式化语言）重复描述同样的信息，或者使用半形式语言和形式化语言描述概要与抽象信息，然后再用自然语言进行详细信息的描述。

## 第16章：

**1.用于需求获取的原型与用于需求验证的原型有何异同？（考过）**

都是原型的一种用法（1分）

用于获取的原型起始于不明确的需求（1分）；侧重于可调整方案（1分）；

用于验证的原型起始于比较明确的需求（1分）；侧重于功能评价方案（1分）；

原型是在软件开发中被广泛使用的一种工具，在软件开发过程中的各个阶段包括需求开发，都会使用不同类型的原型达到不同的目的

当有些需求涉及复杂的动态行为时，可能就需要使用模型来加以验证。面对不确定的需求的时候，需求获取的时候就会使用原型的方法。相同点就是都使用了原型的方法。 区别：原型验证是开始于确定完整的需求，都是比较明确的需求，对于修改的要求不高，但是对性能捕获上要求比较高，由于开始于清晰地需求，所以原型验证更倾向于实验式。而需求获取的原型，一般都是开始于不确定的需求，修改性比较大。并且一般通过原型进行获取的需求，一般不用原型检验

**2多种需求验证的方法应该如何结合运用？**

需求验证的方法：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

每个需求都需要经过评审，对于动态行为评审不能完成的就要通过原型和模拟的方法来验证。在正常的工作当中，可以顺便用上用户手册，测试用例，跟踪等方法在一些错误之处或者一些需求上进行验证，也是比较有效的。总而言之，大多数情况下，需求都是在静态的方式下被加以验证的（评审的方法），也可以说几乎说的需求都要经过评审的方法进行验证，个别动态复杂的需求需要用原型与模拟的方法进行验证，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，这样可以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时很多现代化软件的开发技术的主题，对此现象你有什么看法？你将怎样做可以控制一个需求多变的项目？（考过）**

1. 认识到需求变化是不可避免的；1分
   1. 给出需求变化不可避免原因的解释；2分

包括：问题发生了改变。环境发生了改变。需求基线存在缺陷。

1. 明确需求的变化是需要控制的；1分
   1. 能够描述需求基线的概念；1分

需求基线就是被明确和固定的需求集合，是项目团队在某一特定产品版本中实现的特征和需求集合；

IEEE的定义为：已经通过正式评审和批准的规格说明或产品，它可以作为进一步开发的基础，并且只有通过正式的变更控制过程才能修改它。

在响应需求变更的过程中，项目团队还要及时准确的维护需求基线，审计变更记录：要更新需求基线，保证项目涉众可以访问到最新的需求；

* 1. 能够描述需求变更的过程；2分



1. 明确有的需求变化是需要特殊处理的，要灵活应对变更请求；1分
   1. 能够大概写出特殊处理情况的；2分

包括：

* 推迟产品的交付时间。
* 要求增派人手。当然，这个做法只有在有限的情况下有效，因为很多情况下，增加人手只会使得项目更加落后。
* 要求员工加班工作。一段时期的加班会耗尽员工的储备精力，因此加班不能是长期的，一般以30天为限，否则会产生很多消极影响。因此，这个做法也只能适度的使用。
* 推迟或者去除尚未实现的优先级较低的需求。

容许产品质量的降低。当然，这个做法是最不提倡的，因为低质量的产品会伤害整个开发团队。所以，除非其他的做法都不能达到效果，否则不要使用这种做法。