

1.P3

在软件开发领域，系统模型分为两大类，一类称为**概念模型**，描述了系统是什么，另一类统称为**软件模型**，描述了实现概念模型的软件解决方案，软件模型又可以进一步分为**设计模型**，**实现模型**和**部署模型**等。

2.P26

质量属性的指标是

**可靠性**：是指软件系统在指定环境中没有失败而正常运行的概率

**存活性**：是指当系统的某一部分不能运行时，该软件继续运行或支持关键功能的可能性

**可维护性**：是指发现并改正一个软件故障或对特定的范围进行修改所要求的平均工作

**用户友好性**：是指学习和使用一个软件系统的容易程度

3.P57

**内容耦合**是最高程度的耦合，应该尽量避免使用

如果模块间必须存在耦合，就尽量使用**数据耦合**，少用**控制耦合**，限制**公共耦合**的范围，尽量避免使用**内容耦合**

4.P45

总体设计的工具是

**Yourdon** 提出的模块结构图

层次图

**HIPO** 图

5.P92

为了表达关联的含义，UML 采用以下途径

关联名

导航

角色

可见性

多重性

限定符

聚合

组合

关联类

约束

6.P94

泛化是一般性类目（称为超类或父类）和它的较为特殊性类目（称为子类）之间的一种关系，有时称为“is-a-kind-of”关系

7.P101

一类结构图，用于表达系统或系统成分的静态结构迷性，给出系统或系统成分的一些说明性信息，一类是行为图，用于表达系统或系统成分的动态结构模型，给出系统或系统成分的一些行为模型

类图，对象图，构件图，包图，部署图和组合结构图属于结构图

**Usecase** 图，活动图，状态图，交互图，顺序图，通信图，交互概观图，定时图属于行为图

8.P117

类图可用于创建系统的结构模型

**UseCase** 图可用于创建有关系统的功能模型

状态图可用于创建有关系统的行为生存周期模型

顺序图可用于创建有关系统的交互模型

9.P167

输入为设计类，接口，活动为实现类，输出为构件

输入为构件，接口，活动为完成单元测试，输出为构件

10.P177

白盒测试技术依据是程序的逻辑结构，而黑盒测试结束依据的是软件行为的描述

11.P178

在路径测试中，路径覆盖度量是最强的，一般不可能实现的

语句覆盖是一种最低的测试度量

12.P187

功能测试技术：

等价类测试

边界值测试

因果图测试

13.P203 过程组织有以下过程：

管理过程

基础设施过程

培训过程

改进过程

14.

P255

持续优化过程是过程制度化的最高水平

15.P225

演化模型具有的特性

如果在一次迭代中，有的需求不能满足用户的需求，可再下一次迭代

中予以修正

用户会以变更要求的方式提出反馈，以支持以后增量的需求开发

在演化模型的使用中，即使很好地理解了需求或设计，也很容易弱化

需求分析阶段的工作

考虑需求源的重要性和风险

16.P24

整个需求的主体是**功能需求**

17.P50

总体设计分为 3 个阶段

第一阶段为初始设计

第二阶段为精化设计

第三阶段为复审阶段

18.P58 如果一个模块的各成分之间基本不存在任何关系，则称为**偶然内聚**

几个逻辑上相关的功能被放在同一个模块中，则称为**逻辑内聚**

如果一个模块完成的功能必须在同一时间内执行，但这些功能只是因为时间因素关联在一起，则称为**时间内聚**

如果一个模块内部放入处理成分是相关的，而且这些处理必须以特定的次序执行，则称为**过程内聚**

如果一个模块的所有成分都操作同一个数据集或生成同一个数据集，则称为**通信内聚**

如果一个模块的各个成分和同一个功能密切相关，而且一个成分的输出作为另一个成分的输入，则称为**顺序内聚**

最理想的内聚是**功能内聚**，模块的所有成分对于完成单一的功能都是基本的

19.P60 **宽度**是指同一个层次上模块总数的最大值

**模块扇出**是指一个模块直接控制（调用）的下级模块数目

20.P125 在 RUP 中,规定了 4 个开发阶段

初始阶段，精化阶段，构造阶段，移交阶段

21.

P175 评审，走查和形式化证明等是静态评估技术

**软件测试**是一种常用的动态评估技术

22.P177

白盒测试技术，又称为**结构测试技术**

黑盒测试技术，又称为**功能测试技术**



23.P177 路径测试主要有  
路径覆盖，语句覆盖，分支覆盖，条件覆盖与条件组合覆盖

24.P226 螺旋模型是在瀑布模型和演化模型的基础上，加入两者所忽略的风险分析所建立的一种软件开发模型

25.P228 喷泉模型主要用于支持面向对象技术的软件开发

26.P244CMMI 能力等级

0 级：未完成级

1 级：已执行级

2 级：已管理级

3 级：已定义级

4 级：已定量管理级

5 级：持续优化级

27.P23 软件需求以一种技术形式，描述了一个产品、系统应该具有的功能，性能和其他性质。可见，软件需求是产品、系统设计、实现以及验证的基本信息源之一，是任何软件工程项目的基础。

28.P44

需求阶段的主要任务是完整地定义问题，确定系统的功能和能力，为此，该阶段主要任务一般包括需求发现，需求分析，和需求验证，最终形成系统的软件需求规格说明书

29.P47HIPO 图由 **H** 图和 **IPO** 图两部分组成的

30.P45 结构化设计又进一步分为总体设计和详细设计，其总体设计的目标是建立系统的模块结构，而详细设计要提供实现该模块过程或功能的具体算法

31.P106 参与者大体分为三种

一是使用系统的人

二是与系统交互的其他系统和设备

三是管理，维护系统的人

32.P123

**RUP** 是基于 **UML** 的一种过程框架

RUP 给出的是一种过程指导，与 UML 一起才称得上是一种面向对象开发方法学

33.P126

在 RUP 的每次迭代中都要经历一个核心工作流，即需求获取，分析，设计，实现和测试

34.P177 白盒测试技术典型的是路径测试技术，依据的是程序的逻辑结构

黑盒测试技术包括事物处理流程技术，状态测试技术，定义域测试技术等

35.P191

单元测试主要检验软件设计的最小单元-模块，该测试以详细设计文档为指导，测试模块内的重要控制路径，一般来说，单元测试往往采用白盒测试技术

36.P199 按过程主体把软件生存周期过程分为 3 类

基本过程

支持过程

组织过程

37.P282

集成了 3 个源模型

软件 **CMM**

系统工程 **CMM**

集成产品开发 **CMM**

38.P282

CMMI 提供了两种改善路径，一是称为能力等级的过程改善路径，该路径可使组织针对单一过程域，不断改善该过程域，二是称为成熟度等级的过程改善路径，该路径可使组织通过关注一组过程域，不断改善一组相关的过程域。

39.P17

关于如何实现这样的映射，这是技术层面上的问题，这一问题又涉及两方面的内容一是过程方向，二是过程途径

#### 40.P26 需求发现技术 5 个指标

自悟存在的风险：无法验证发现的需求是否满足用户的要求，无法验证发现的需求是不是正确的

交谈存在的风险：在交谈器件所获得的需求可能不断增长，或是以前没有认识到的合理需求的一种表现，或是“完美蠕行”病症的体现，以至于很难控制，可能导致超出项目成本和进度的限制

观察存在的风险：

客户可能抵触这一观察，其原因是他们认为开发者打扰了他们的正常业务

客户可能认为开发者在签约之前，就已经熟悉了他们的业务

小组会存在的风险：

如果会议组织不到位或受到某些客观环境的限制，就有可能过多地召开这样的会议，并产生一些相互矛盾的需求

提炼存在的风险：

与自悟一样，无法验证所发现的需求是否满足用户的要求，无法验证发现的需求是否正确

#### 41.P37

需求分析的首要任务是建立系统功能模型，为此结构化分析方法给出了一种表达功能模型的工具，即数据流图

#### 42.P102 类图是构件图和部署图的基础

#### 43.P169

其中，UML 作为一种可视化建模语言，给出了表达事物和事物之间关系的基本术语，给出了多种模型的表达工具，而 RUP 利用这些术语定义了需求获取层，系统分析层，设计层，实现层，并给出了实现各层模型之间映射的基本活动以及相关的指导

#### 44.P192



有效性测试的目标是发现软件实现的功能与需求规格说明书不一致的错误

45.P246

达到共用目标 1 的“已执行过程”是制度化程度最低的过程，而通过过程改进，

达到共用目标 5“持续优化过程”是制度化程度最高的过程

46.P78 属性和操作的可见性是否可以被其他类所使用

操作是对外提供的一种服务

所以，面向对象技术的封装的特性

47.P19

软件开发的本质，即实现问题空间的概念和处理逻辑到解空间的概念和处理逻辑之间的映

射，概括地介绍了实现这一映射的基本途径，即系统建模

所谓系统建模，是指运用所掌握的知识，通过抽象，给出该系统的一个结构-系统模型

48.P25

接口需求可分为以下主要 7 类

用户接口

硬件接口

软件接口

通信接口

内存接口

内存约束

运行

地点需求

设计约束需要考虑以下各方面的问题

法规政策

硬件限制

与其他应用的接口

并发操作



审计功能

控制功能

高级语言需求

握手协议

应用的关键程度

安全和保密

49.P57 内容耦合是最高程度的耦合

数据耦合是最低的一种耦合形式

50.P61

高内聚低耦合这一模块独立性原则

51.P64

详细设计工具有：

程序流程图

盒图

**PAD 图**

类程序设计语言

52.P92

显然聚合是关联的一种特殊形式，表达的是一种“整体、部分”关系

53.P168

理解表 5-9 中的测试活动内容

54.P199

基本过程的 5 个过程是

获取过程

供应过程

开发过程

运行过程

## 维护过程

55.P258

在 CMMI 中，应用于一个组织过程改善的成熟度等级有 5 个

1 级：初始级

2 级：已管理级

3 级：已定义级

4 级：已定量管理级

5 级：持续优化级

56.P61

扇入越大则共享该模块的上级模块数目越多

57.P103

状态图支持系统生存周期的建模

58.P169

RUP 框架体现了 3 个突出特征：

用况驱动

体系结构为中心以及迭代

增量式开发

59.P192

自顶向下的集成测试是一种递增组装软件的方法，从主控模块开始，沿控制层次向下，先深度或先宽度地将模块逐一组合起来，形成与设计相符的软件结构

60.P222

瀑布模型规定了各开发阶段的活动：

系统需求，软件需求，需求分析，设计，编码，测试和运行，并且自上而下具有相互衔接的固定顺序，还规定了每一阶段的输入，即工作对象以及本阶段的工作成果，作为输出传送到下一阶段

61.P282

CMMI 模型基于过程途径思想，通过过程把软件质量的 3 个支撑点，受训的人员，

规程和方法，工具和设备进行集成，以开发所期望的系统，产品

62.P19

在需求层上创建系统概念模型是对事物系统的抽象

63.P29

特定需求是文档的技术核心

64.P36

加工一般采用动宾结构

65.P99

为了控制信息组织的复杂性，UML 提供了组织信息的一种通用机制-包

66.P221

喷泉模型体现了软件创建所固有的迭代和无间隙的特征

67.P15

20 世纪 60 年代以来，随着计算机的广泛使用，软件生产率，软件质量远远满足不了社会发展的需求，成为社会，经济发展的制约因素，人们通常把这一现象称为“软件危机”

68.P56

基于模块“高内聚低耦合”的原则，提高模块的独立性

模块通常由两部分组成：

接口

模块体

69.P109

UML 把状态分为 3 类：即初态，终态和通常状态

70.P92

组合又是聚合的一种特殊形式，如果在一个时间段内，整体类的实例中至少包含一个

个部分类的实例，并且该整体类的实例负责创建和消除部分类的实例，特别是如果整体类

的实例和部分类的实例具有相同的生存周期，那么这样的聚合称为组合

71.P175

测试的目标只能是发现错误

72.P193

有效性测试通常采用黑盒测试技术



单元测试集中于单个模块的功能和结构检验

集成测试集中于模块组合的功能和软件结构检验

有效性测试验证软件需求的可追溯性

系统测试验证将软件融于更大系统中时整个系统的有效性

73.P192

集成测试目标是发现与接口有关的错误