



软件架构设计-必须掌握的知识点



软件架构风格

- 是指描述特定软件系统组织方式的惯用模式。组织方式描述了系统的组成构件和这些构件的组织方式，惯用模式则反映众多系统共有的结构和语义。



架构风险

- 是指架构设计中潜在的、存在问题的架构决策所带来的隐患。



风险点与非风险点

- 不是以标准专业术语形式出现的，只是一个常规概念，即可能引起风险的因素，可称为风险点。某个做法如果有隐患，有可能导致一些问题，则为风险点;而如果某件事是可行的可接受的，则为非风险点。



质量属性和效用树

- 有哪些质量属性，这些质量属性分别代表的含义以及其设计策略。



敏感点和权衡点

- 敏感点：指为了实现某种特定的质量属性，一个或多个构件所具有的特性。
- 权衡点：影响多个质量属性的特性，是多个质量属性的敏感点。



典型的架构

- 常见的MVC架构、J2EE架构、SOA架构等



软件架构设计-质量属性效用树、质量属性



1. 性能：指系统的响应能力，即要经过多长时间才能对某个事件做出响应，或者在某段时间内系统所能处理的事件的个数。如响应时间、吞吐量。

- 设计策略：优先级队列、增加计算资源、减少计算开销、引入并发机制、采用资源调度等。



2. 可靠性：是指系统在一段时间内保持正常运行而不发生故障的能力。它强调了系统的稳定性和可靠性，通常是通过衡量系统在一段时间内发生故障的概率来评估的。一个可靠性高的系统意味着它很少出现故障，用户可以信任它的稳定性。可靠性有一些指标需要了解。如MTTF(平均故障时间)、MTBF(平均故障间隔时间)、MTTR(平均故障修复时间)。

- 设计策略：心跳、Ping/Echo、冗余、选举。



3. 可用性：是指系统在需要的时候可供使用的能力，即系统处于可操作状态的时间比例。可用性通常通过计算系统在一定时间内可操作的百分比来评估。一个高可用性的系统意味着它在大部分时间内都是可用的，用户可以随时访问和使用它。

- 设计策略：心跳、Ping/Echo、冗余、选举。



软件架构设计-质量属性效用树、质量属性



可靠性和可用性的区别：假设有两家云存储服务提供商：Provider A 和 Provider B。

- Provider A 的服务在过去一年中从未发生过故障，用户可以随时访问和上传文件。这表明 Provider A 的系统具有很高的可靠性，因为它极少出现故障。
- Provider B 的服务在过去一年中总共停机了5小时，但它提供了冗余备份和快速恢复机制，因此用户在服务中断后很快就可以继续使用。这表明 Provider B 的系统具有很高的可用性，因为它在大部分时间内都是可操作的，尽管它发生了一些故障。
- 在这个示例中，Provider A 的系统更可靠，因为它几乎没有发生故障。Provider B 的系统更可用，因为它尽管发生了一些故障，但它能够快速恢复，以确保用户可以继续使用。**可靠性强调系统不发生故障，而可用性强调系统在需要时可供使用。两者都是衡量系统性能和质量的重要标准，但它们关注的方面略有不同。**



4. 安全性：**是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。如保密性、完整性、不可抵赖性、可控性。**

- 设计策略：**入侵检测、用户认证、用户授权、追踪审计。**



软件架构设计-质量属性效用树、质量属性



5. 可修改性：指能够快速以较高的性能价格比对系统进行变更的能力。通常以某些具体的变更为基准，通过考察这些变更的代价衡量。

- 设计策略：接口-实现分类、抽象、信息隐藏（是不是感觉有点像结构化开发和面向对象开发的设计原则）。



6. 功能性：是系统所能完成所期望的工作的能力。一项任务的完成需要系统中许多或大多数构件的相互协作。



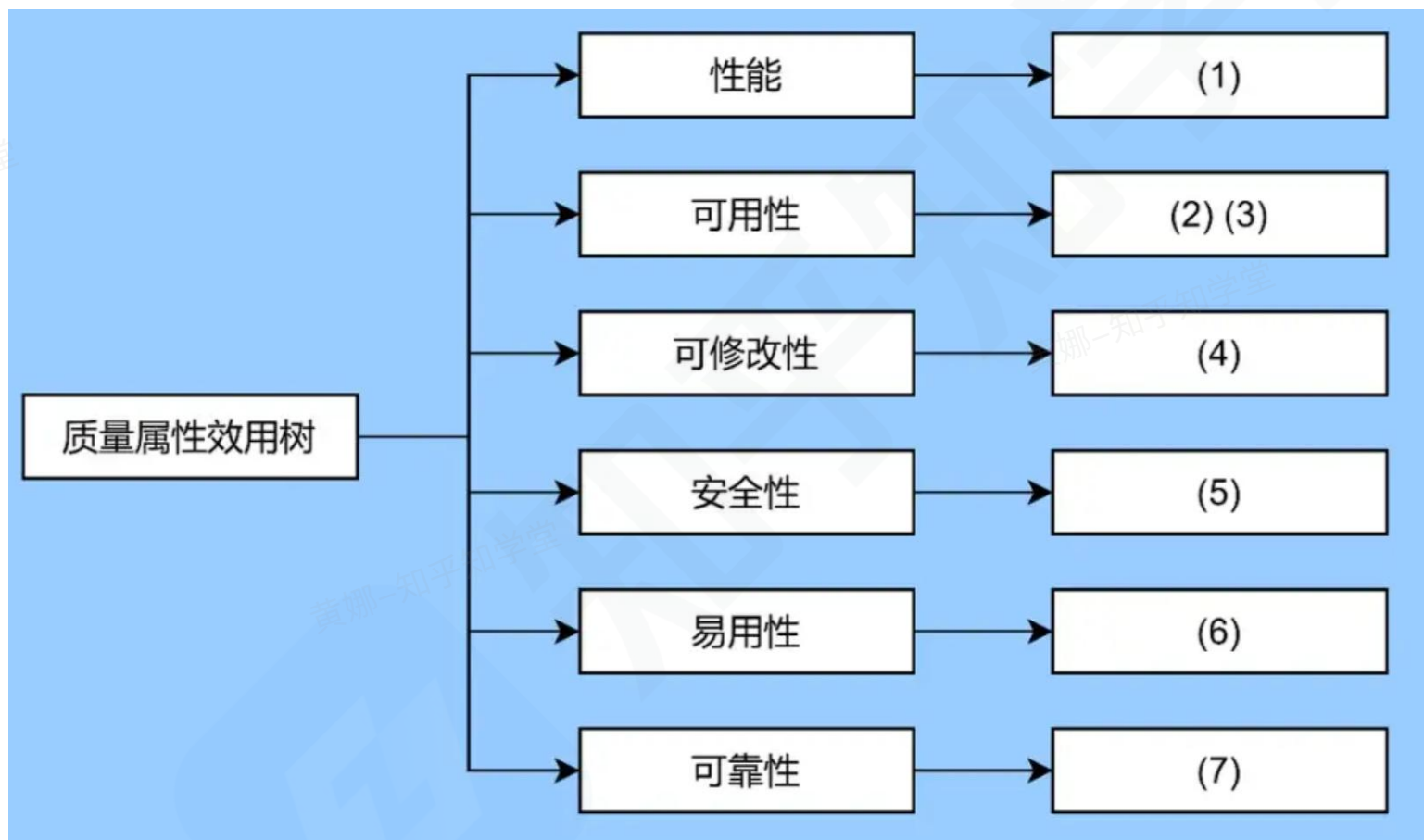
7. 可变性（可扩展）：指体系结构经扩充或变更而成为新体系结构的能力。这种新体系结构应该符合预先定义的规则，在某些具体方面不同于原有的体系结构。当要将某个体系结构作为一系列相关产品的基础时，可变性是很重要的。



8. 互操作性：作为系统组成部分的软件不是独立存在的，经常与其他系统或自身环境相互作用。为了支持互操作性，软件体系结构必须为外部可视的功能特性和数据结构提供精心设计的软件入口。程序和用其他编程语言编写的软件系统的交互作用就是互操作性的问题，也影响应用的软件体系结构。



软件架构设计-质量属性效用树、质量属性





质量属性真题详解

请详细阅读有关软件架构评估方面的说明，在答题纸上回答问题1至问题2。

【说明】某电子商务公司拟升级目前正在使用的在线交易系统，以提高客户网上购物时在线支付环节的效率和安全性。公司研发部门在需求分析的基础上，给出了在线交易系统的架构设计。公司组织相关人员召开了针对架构设计的评估会议，会上用户提出的需求、架构师识别的关键质量属性场景和评估专家的意见等内容部分列举如下：

- (a)在正常负载情况下，系统必须在0.5秒内响应用户的交易请求；
- (b)用户的信用卡支付必须保证99.999%的安全性；
- (c)系统升级后用户名要求至少包含8个字符；
- (d)网络失效后，系统需要在2分钟内发现错误并启用备用系统；
- (e)在高峰负载情况下，用户发起支付请求后系统必须在10秒内完成支付功能；
- (f)系统拟采用新的加密算法，这会提高系统安全性，但同时会降低系统的性能；
- (g)对交易请求处理时间的要求将影响系统数据传输协议和交易处理过程的设计；
- (h)需要在30人月内为系统添加公司新购买的事务处理中间件；
- (i)现有架构设计中的支付部分与第三方支付平台紧耦合，当系统需要支持新的支付平台时，这种设计会导致支付部分代码的修改，影响系统的可修改性；
- (j)主站点断电后，需要在3秒内将访问请求重定向到备用站点；
- (k)用户信息数据库授权必须保证99.999%可用；
- (l)系统需要对Web界面风格进行修改，修改工作必须在4人月内完成；
- (m)系统需要为后端工程师提供远程调试接口，并支持远程调试。

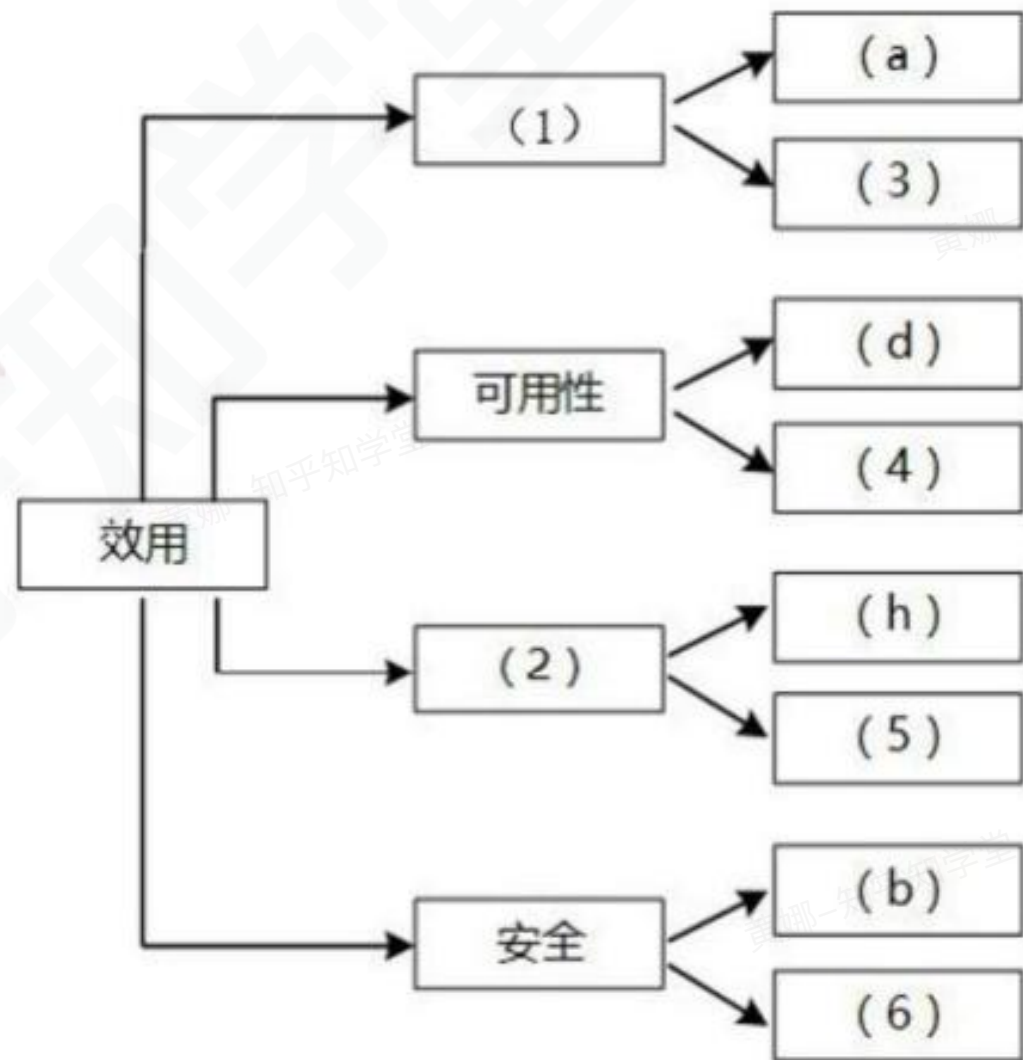


质量属性真题详解

【问题1】(12分)

在架构评估过程中，质量属性效用树(utility tree)是对系统质量属性进行识别和优先级排序的重要工具。请给出合适的质量属性，填入图4-1中(1)、(2)空白处；并选择题干描述的(a)~(m)，填入(3)~(6)空白处，完成该系统的效用树。

【问题2】(13分)在架构评估过程中，需要正确识别系统的架构风险、敏感点和权衡点，并进行合理的架构决策。请用300字以内的文字给出系统架构风险、敏感点和权衡点的定义，并从题干(a)~(m)中各选出1个对系统架构风险、敏感点和权衡点最为恰当的描述。



质量属性真题详解

试题答案

【问题1】(12分)

(1)性能

(2)可修改性

(3)(e)

(4)(j)

(5)(l)

(6)(k)

【问题2】(13分)

系统架构风险是指架构设计中潜在的、存在问题的架构决策所带来的隐患。

敏感点是指为了实现某种特定的质量属性，一个或多个构件所具有的特性。

权衡点是影响多个质量属性的特性，是多个质量属性的敏感点。

风险点: (g)

敏感点: (i)

权衡点: (f)