

NEW PLAN

软考高级架构师

• 一段新征程 •





软件系统属性



软件系统属性包括**功能属性**和**质量属性**，软件架构重点关注的是质量属性。



架构的基本需求是在**满足功能属性的前提下，关注软件系统质量属性**。为了精确、定量地表达系统的质量属性，通常会采用**质量属性场景**的方式进行描述。在确定软件系统架构，精确描述质量属性场景后，就需要对**系统架构进行评估**。软件系统架构评估是在对架构分析、评估的基础上，**对架构策略的选取进行决策**。它也可以灵活地运用于软件架构评审等工作。



软件系统质量属性 (Quality Attribute) 是一个系统的可测量或者可测试的属性，用来描述系统满足利益相关者需求的程度。简单来说，**软件系统质量属性是指软件系统满足用户需求的程度，是衡量软件系统质量的重要指标**。基于软件系统的生命周期，可以将软件系统的质量属性分为**开发期质量属性**和**运行期质量属性**2个部分



软件系统的质量属性



开发期质量属性主要指在**软件开发阶段所关注的质量属性**，这个阶段关注人群主要是开发者，包含**6**个方面。

- 易理解性：指设计被开发人员理解的难易程度。
- 可扩展性：软件因适应新需求或需求变化而增加新功能的能力，也称为灵活性。
- 可重用性：指重用软件系统或某一部分的难易程度。
- 可测试性：对软件测试以证明其满足需求规范的难易程度。
- 可维护性：当需要修改缺陷、增加功能、提高质量属性时，识别修改点并实施修改的难易程度。
- 可移植性：将软件系统从一个运行环境转移到另一个不同的运行环境的难易程度。



运行期质量属性主要指在**软件运行阶段所关注的质量属性**，这个阶段关注人群主要是用户，包含**7**个方面。

- 性能：性能是指软件系统及时提供相应服务的能力，如速度、吞吐量和容量等的要求。
- 安全性：指软件系统同时兼顾向合法用户提供服务，以及阻止非授权使用的能力。
- 可伸缩性：指当用户数和数据量增加时，软件系统维持高服务质量的能力。例如，通过增加服务器来提高能力。
- 互操作性：指本软件系统与其他系统交换数据和相互调用服务的难易程度。
- 可靠性：软件系统在一定的时间内持续无故障运行的能力。
- 可用性：指系统在一定时间内正常工作的时间所占的比例。可用性会受到系统错误，恶意攻击，高负载等问题的影响。
- 鲁棒性：是指软件系统在非正常情况(用户进行了非法操作、相关软硬件系统发生了故障)下仍能够正常运行的能力，也称健壮性或容错性。



面向架构评估的质量属性



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(平均无故障时间)、MTBF(平均故障间隔时间)、MTTR(平均故障修复时间)。MTBF=MTTF+MTTR

- 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。



面向架构评估的质量属性



可靠性和可用性的区别：假设有两家云存储服务提供商：A公司和B公司。

- A公司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A公司的系统具有很高的可靠性，因为它极少出现故障。
- B公司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B公司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，A公司的系统更可靠，因为它几乎没有发生故障。B公司的系统更可用，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。



面向架构评估的质量属性



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。



9. 易用性：它关注的是软件系统的用户界面和交互设计，以确保用户能够轻松、高效地使用系统，并感到满意。易用性不仅关乎用户界面的外观，还包括用户体验、交互流程和用户学习曲线等方面



质量属性场景描述



为了精确描述软件系统的质量属性，通常采用质量属性场景 (Quality Attribute Scenario) 作为描述质量属性的手段。质量属性场景是一个具体的质量属性需求，是利益相关者与系统的交互的简短陈述。



质量属性场景是一种用于描述系统如何满足特定质量属性需求的情境或情景。它由6部分组成：

- 刺激源(谁)：这是某个生成该刺激的实体(人、计算机系统或者任何其他刺激器)。
- 刺激(做什么)：该刺激是当刺激到达系统时需要考虑的条件。
- 环境(在什么样的环境下)：该刺激在某些条件内发生。当激励发生时，系统可能处于过载、运行或者其他情况。
- 制品(对哪个功能)：某个制品被激励。这可能是整个系统，也可能是系统的一部分。
- 响应(得到什么反馈)：该响应是在激励到达后所采取的行动。
- 响应度量(对反馈进行度量)：当响应发生时，应当能够以某种方式对其进行度量，以对需求进行测试。



质量属性场景描述



举例：假设我们正在设计一个电子商务网站，其中一个关键的质量属性是性能，希望确保网站在高负载时仍能快速响应：



场景：高负载购物日：



描述：在每年的假期购物季节（如圣诞节），我们预期会有大量的用户访问我们的网站，进行在线购物。在高负载购物日，我们预计用户流量将增加至平时的**10倍**。

- 刺激源（谁）：刺激源是购物网站的用户，这些用户在特定的购物季节（例如圣诞节）涌入网站，导致高负载情况。
- 刺激（做什么）：刺激是用户访问购物网站，浏览产品、添加商品到购物车以及进行结算等在线购物活动。
- 环境（在什么样的环境下）：环境是购物网站在特定的日期范围内，即12月24日至12月25日，而且用户流量显著增加，购物活动频繁进行。
- 制品（对哪个功能）：制品是购物网站的整个系统，特别是与用户交互的部分，包括网站的前端和后端。
- 响应（得到什么反馈）：响应是购物网站采取的一系列行动，以确保在高负载购物日仍能够提供快速响应和正常运行的服务。这包括使用负载均衡、缓存静态内容、垂直扩展和水平扩展等措施。
- 响应度量（对反馈进行度量）：响应度量是在高负载购物日期间对系统性能进行度量的过程。它包括测量网页加载时间、服务器资源利用率、响应时间、错误率等指标，以确保系统在性能方面达到了预期目标。



某公司欲开发一个在线交易网站，在架构设计阶段，公司的架构师识别出3个核心质量属性场景。其中"网站正常运行时，用户发起的交易请求应该在3秒内完成"主要与(58)质量属性相关，通常可采用(59)架构策略实现该属性："在线交易主站宕机后，能够在3秒内自动切换至备用站点并恢复正常运行"主要与(60)质量属性相关，通常可采用(61)架构策略实现该属性："系统应该具备一定的安全保护措施，从而能够抵挡恶意的入侵破坏行为，并对所有针对网站的攻击行为进行报警和记录"主要与(62)质量属性相关，通常可采用(63)架构策略实现该属性。

(58)A.可用性

B.性能

C.易用性

D.可修改性

(59)A.抽象接口

B.信息隐藏

C.主动冗余

D.资源调度

(60)A.可测试性

B.易用性

C.可用性

D.互操作性

(61)A.记录/回放

B.操作串行化

C.心跳

D.增加计算资源

(62)A.可用性

B.安全性

C.可测试性

D.可修改性

(63)A.追踪审计

B.Ping/Echo

C.选举

D.维护现有接口



考试真题



【2022】54-55.软件系统质量属性(Quality Attribute)是一个系统的可测量或者可测试的属性，它被用来描述系统满足利益相关者需求的程度，其中，()关注的是当需要修改缺陷、增加功能、提高质量属性时，定位修改点并实施修改的难易程度；()关注的是当用户数和数据量增加时，软件系统维持高服务质量的能力。

A.可靠性

B.可测试性

C.可维护性

D.可重用性

A.可用性

B.可扩展性

C.可伸缩性

D.可移植性



考试真题



【2022】56-57.为了精确描述软件系统的质量属性，通常采用质量属性场景作为描述质量属性的手段。质量属性场景是一个具体的质量属性需求，是利益相关者与系统的交互的简短陈述，它由刺激源、刺激、环境、制品、()六部分组成。其中，想要学习系统特性、有效使用系统、使错误的影响最低、适配系统、对系统满意属于()质量属性场景的刺激。

A.响应和响应度量 B.系统和系统响应 C.依赖和响应 D.响应和优先级

A.可用性 B.性能 C.易用性 D.安全性



考试真题



【2022】60.平均失效等待时间(mean time to failure,MTTF)和平均失效间隔时间(mean time between failure,MTBF)是进行系统可靠性分析时的重要指标,在失效率为常数和修复时间很短的情况下,()。

A.MTTF远远小于MTBF

B.MTTF和MTBF无法计算

C.MTTF远远大于MTBF

D.MTTF和MTBF几乎相等



考试真题



【2022】61-62.在进行软件系统安全性分析时，()保证信息不泄露给未授权的用户、实体或过程；完整性保证信息的完整和准确，防止信息被非法修改；()保证对信息的传播及内容具有控制的能力，防止为非法者所用。

A.完整性

B.不可否认性

C.可控性

D.机密性

A.完整性

B.安全审计

C.加密性

D.可控性



系统架构评估



敏感点：是指为了实现某一种特定的质量属性，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。



风险承担者或者称为利益相关人：系统的架构涉及很多人的利益，这些人都对架构施加各种影响，以保证自己的目标能够实现



软件架构评估在架构设计之后，系统设计之前，因此与设计、实现、测试都没有关系。评估的目的是为了评估所采用的架构是否能解决软件系统需求，但不是单纯的确定是否满足需求，有时候还需要针对质量属性进行评估架构是否能满足。

表8-7 系统架构评估涉及的问题

风险承担者	职责	所关心的问题
系统生产者		
软件系统架构师	负责软件架构的质量需求间进行权衡的人	对其他风险承担者提出的质量需求的折中和调停
开发人员	设计人员或程序员	架构描述的清晰与完整、各部分的内聚性与受限耦合、清楚的交互机制
维护人员	系统初次部署完成后对系统进行更改的人	可维护性，确定出某个更改发生后必须对系统中哪些地方进行改动的能力
集成人员	负责构件集成和组装的开发人员	与上同
测试人员	负责系统测试的开发人员	集成、一致的错误处理协议，受限的构件耦合、构件的高内聚性、概念完整性
标准专家	负责所开发软件必须满足的标准细节的开发人员	对所关心问题的分离、可修改性和互操作性
性能工程师	分析系统的工作产品以确定系统是否满足其性能及吞吐量需求的人员	易理解性、概念完整性、性能、可靠性
安全专家	负责保证系统满足其安全性需求的人员	安全性
项目经理	负责为各小组配置资源、保证开发进度、保证不超出预算的人员，负责与客户沟通	架构层次清晰，便于组建小组；任务划分结构、进度标志和最后期限等
产品线经理	设想该架构和相关资产怎样在该组织的其他开发中得以重用的人员	可重用性、灵活性



系统架构评估-三种常用的评估方式



1. 基于调查问卷(检查表)的方式：类似于需求获取中的问卷调查方式，只不过是架构方面的问卷，这种方式要求评估人员对领域和架构具有一定的了解，因为他们需要能够理解并回答与架构相关的问题。这种方式通常用于获取主观反馈和意见，以便改进架构。

- 举例：对于一个大型电子商务平台的架构评估，评估团队可以向开发团队发送架构调查问卷，询问他们对性能、可维护性和安全性等方面的看法。问题可能包括：“您认为当前系统的性能是否满足需求？”或者“您认为系统的安全措施是否足够？”



2. 基于度量的方式：制定一些定量指标来度量架构，如代码行数、内存使用、响应时间等，来评估系统的各个方面。这种方式要求评估人员对架构的技术细节和度量标准有一定了解

- 举例：对于一个大型数据库管理系统的架构评估，评估团队可以度量数据库查询的平均响应时间、事务处理的吞吐量、数据库表的大小等指标。这些度量可以用于评估系统的性能和可伸缩性。



系统架构评估-三种常用的评估方式



3. 基于场景的方式：主要方法。这是一种系统性的评估方法，涉及定义一系列场景或情境，以测试系统在各种条件下的表现，每个场景描述了一种特定的使用情况，包括输入、预期输出和性能指标。评估人员会模拟这些场景，并评估系统是否满足质量属性需求。**场景是从风险承担者的角度对与系统交互的简单描述。**

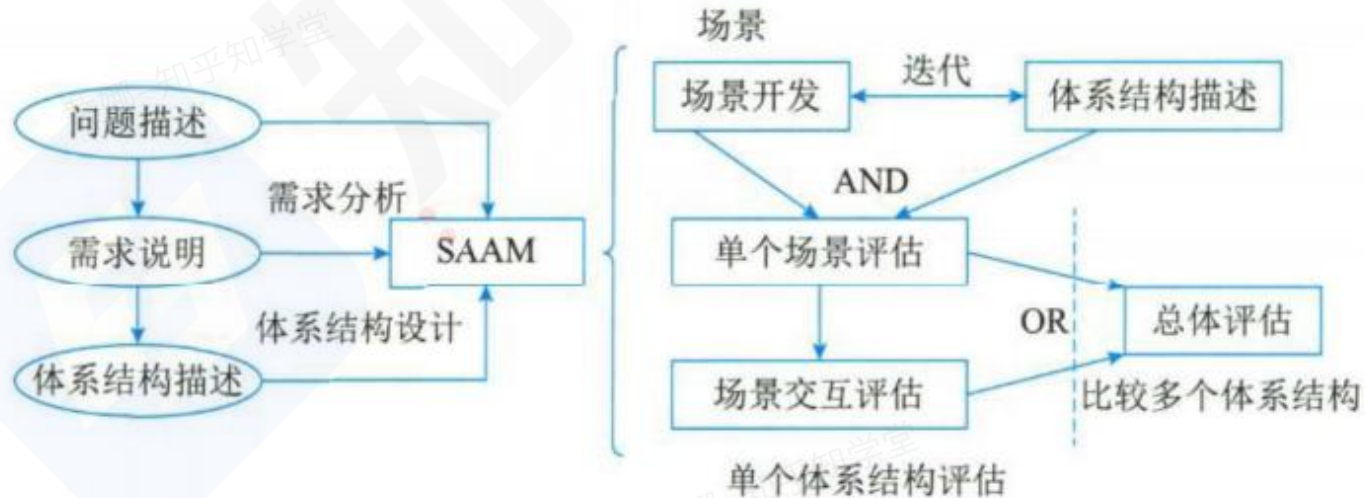
- **实施过程：**首先要确定应用领域的功能和软件架构的结构之间的映射，然后要设计用于体现待评估质量属性的场景(即4+1视图中的场景)，最后分析软件架构对场景的支持程度。要求评估人员即对领域熟悉，也对架构熟悉
- **从三个方面对场景进行设计：**刺激(事件)、环境(事件发生的环境)和响应(架构响应刺激的过程)。
- **举例：**对于一个网络视频流媒体服务的架构评估，评估团队可以定义场景，如“同时1000名用户观看高清视频”或“在网络拥塞时仍能提供无缝的视频流”。然后，他们会模拟这些场景，测试系统的性能、可用性和容错性。

系统架构评估方法



基于场景的架构分析方法SAAM: **SAAM是一种非功能质量属性的架构分析方法**, 是最早形成文档并得到广泛应用的软件架构分析方法。

- 特定目标。SAAM的目标是对描述应用程序属性的文档, 验证基本的架构假设和原则。
- 评估技术。SAAM所使用的评估技术是场景技术。
- 质量属性。这一方法的基本特点是把任何形式的质量属性都具体化为场景, 但可修改性是SAAM分析的主要质量属性。
- 风险承担者。SAAM协调不同参与者之间感兴趣的共同方面, 作为后续决策的基础, 达成对架构的共识。
- 架构描述。SAAM用于架构的最后版本, 但早于详细设计。架构的描述形式应当被所有参与者理解。功能、结构和分配被定义为描述架构的3个主要方面。
- 方法活动。**SAAM的主要输入是问题描述、需求声明和架构描述**。下图描绘了SAAM分析活动的相关输入及评估过程。包括5个步骤, 即**场景开发、架构描述、单个场景评估、场景交互和总体评估**。



系统架构评估方法

架构权衡分析法ATAM，是一种系统架构评估方法，主要在系统开发之前，针对性能、可用性、安全性和可修改性等质量属性进行评价和折中，让架构师明确如何权衡多个质量目标，参与者有评估小组、项目决策者和其他项目相关人。

ATAM被分为四个主要的活动领域,分别是场景和需求收集、体系结构视图和场景实现、属性模型构造和分析以及架构评审与折中 整个评估过程强调以属性(质量属性)作为架构评估的核心概念。

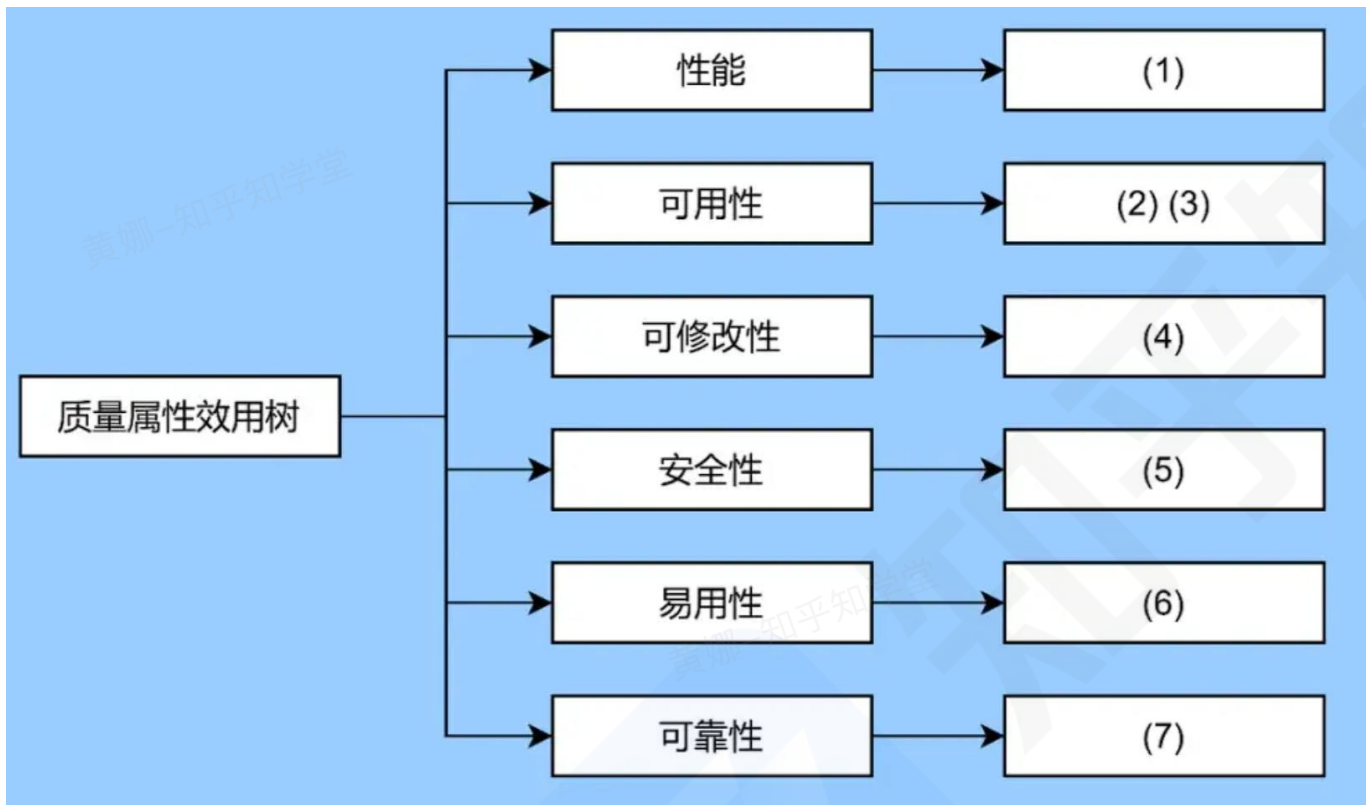
ATAM 方法采用效用树 (Utility tree) 这一工具来对质量属性进行分类和优先级排序。效用树的结构包括：树根—质量属性—属性分类—质量属性场景(叶子节点)。需要注意的是，ATAM主要关注4类质量属性：性能、安全性、可修改性和可用性，这是因为这4个质量属性是利益相关者最为关心的。

得到初始的效用树后，需要修剪这棵树，保留重要场景(通常不超过50个),再对场景按重要性给定优先级(用 H/M/L 的形式),再按场景实现的难易度来确定优先级(用 H/M/L的形式),这样对所选定的每个场景就有一个优先级对(重要度、难易度),如 (H、L) 表示该场景重要且易实现。



系统架构评估方法

在书本的293页有图可以了解。



在需求分析与架构设计阶段，公司提出的需求和质量属性描述如下：

(a)管理员能够在页面上灵活设置折扣力度规则和促销活动逻辑，设置后即可生效；

(b)系统应该具备完整的安全防护措施，支持对恶意攻击行为进行检测与报警；

(c)在正常负载情况下，系统应在0.3秒内对用户的界面操作请求进行响应；

(d)用户名是系统唯一标识，要求以字母开头，由数字和字母组合而成，长度不少于6个字符；

(e)在正常负载情况下，用户支付商品费用后在3秒内确认订单支付信息；

(f)系统主站点电力中断后，应在5秒内将请求重定向到备用站点；

(g)系统支持横向存储扩展，要求在2人天内完成所有的扩展与测试工作；

(h)系统宕机后，需要在10秒内感知错误，并自动启动热备份系统；

(i)系需要内置接口函数，支持开发团队进行功能调试与系统诊断；

(j)系统需要为所有的用户操作行为进行详细记录，便于后期查阅与审计；

(k)支持对系统的外观进行调整和配置，调整工作需要4人天内完成。



系统架构评估方法



ATAM的阶段解释（见书本289-304页）：



1. 描述和介绍阶段：

- 目标：此阶段的目标是收集相关架构材料，定义评估的目标和评估的软件架构，明确要优化的质量属性以及介绍ATAM方法的步骤和原则
- 活动：在这个阶段，评估团队会与项目干系人一起定义评估的目标，确定评估的软件架构，并收集架构文档和相关信息。团队还会介绍ATAM方法的步骤，以确保所有参与者了解评估的过程。
- 示例：对于电子商务网站的架构评估，评估团队会与项目干系人合作，确定评估的目标是提高性能和安全性。他们收集有关网站架构的文档，如架构图和设计文档。



2. 调查和分析阶段：

- 目标：此阶段的目标是确定架构方法，分析架构并评估这些架构方法对质量属性的影响，同时识别潜在的问题和权衡决策
- 活动：在这个阶段，评估团队会定义架构方法，分析不同的架构设计，生成质量属性效应树以表示不同决策对质量属性的影响。他们还会识别潜在的问题和决策权衡。
- 产出：在此阶段，产出质量属性效应树（Quality Attribute Utility Tree），用于表示不同架构决策对质量属性的影响以及它们之间的权衡关系。
- 示例：对于电子商务网站的架构评估，评估团队会定义不同的架构决策，如引入缓存或增加服务器资源。他们生成质量属性效应树，以分析这些决策对性能和安全性的影响。



系统架构评估方法



3. 测试阶段:

- 目标: 测试阶段旨在验证架构是否满足质量属性需求。
- 活动: 团队创建测试用例来模拟质量属性场景, 包括性能测试、安全性测试等。他们运行这些测试用例, 测量系统的性能和行为, 并记录测试结果。在此阶段, 评估团队讨论各种质量属性场景, 对它们进行分级, 以确定哪些场景对系统的关键性最高。团队还会分析不同的架构方法, 以确定哪种方法最有可能满足关键场景的需求。最后, 项目干系人会对不同的架构方法和场景分级进行投票, 以帮助团队确定最佳的架构方案。
- 示例: 评估团队可能会讨论电子商务网站的性能、可伸缩性和安全性场景, 分级它们的重要性, 并分析引入缓存或增加服务器资源等不同架构方法的影响。项目干系人会进行投票, 以选择最合适的架构方法。



4. 报告阶段:

- 目标: 报告阶段的目标是总结评估的结果、提供改进建议, 并为决策者提供决策依据
- 活动: 团队生成ATAM评估报告, 其中包括评估的发现、性能数据、可能的改进建议以及权衡决策。报告应该清晰地传达关键信息, 以便决策者可以做出明智的架构决策。
- 示例: 电子商务网站的评估报告可能包括性能测试结果、安全性评估、建议的架构改进, 以及与质量属性场景相关的权衡决策。报告将提供给项目管理团队, 以指导后续的架构决策和改进。

系统架构评估方法





系统架构评估方法



成本效益分析法CBAM: 用来对架构建立的成本来进行设计和建模，让决策者根据投资收益率来选择合适的架构，可以看做对ATAM的补充，在ATAM确定质量合理的基础上，再对效益进行分析。



有下列步骤:

- 整理场景(确定场景，并确定优先级，选择三分之一优先级最高的场景进行分析);
- 对场景进行细化(对每个场景详细分析，确定最好、最坏的情况);
- 确定场景的优先级(项目干系人对场景投票，根据投票结果确定优先级);
- 分配效用(对场景响应级别确定效用表，建立策略、场景、响应级别的表格);
- 形成“策略-场景-响应级别的对应关系”;
- 确定期望的质量属性响应级别的效用(根据效用表确定所对应的具体场景的效用表);
- 计算各架构策略的总收益;
- 根据受成本限制影响的投资报酬率选择架构策略(估算成本，用上一步的收益减去成本，得出收益，并选择收益最高的架构策略)。



系统架构评估方法



其他评估方法(见书本283页-288页, 知道有这些东西即可)



1. SAEM方法。将软件架构看作一个最终产品以及设计过程中的一个中间产品, 从外部质量属性和内部质量属性两个角度来阐述它的评估模型, 旨在为软件架构的质量评估创建一个基础框架。外部属性指用户定义的质量属性, 而内部属性指开发者决定的质量属性。该软件架构评估模型包含以下几个流程。

- 对待评估的质量属性进行规约建模。
- 为外部和内部的质量属性创建度量准则, 先从评估目的(如软件架构比较、最终产品的质量预测), 评估角度(如开发者、用户、维护者), 评估环境(架构作为最终产品或设计中间产品)出发来定义架构评估的目标, 再根据目标相关的属性来提出问题, 然后回答每个问题并提出相应的度量准则。
- 评估质量属性, 包括数据收集、度量和结果分析3个活动。



2. SAABNet方法。是一种用来表达和使用定性知识以辅助架构的定性评估。该方法来源于人工智能允许不确定、不完整知识的推理。该方法使用BBN来表示和使用开发过程中的知识, 包含定性和定量的描述, 其中定性的描述是所有结点的图, 定量的描述是每个结点状态相关的条件概率。其中的变量可分为3类, 即架构质量属性变量(如可维护性、灵活性等)、质量属性的度量准则变量(如容错性、响应性等)和架构特征变量(如继承深度、编程语言等), 高层抽象的质量属性变量分解为低层抽象的度量准则变量, 度量准则变量则分解为更低层抽象的架构特征变量。



系统架构评估方法



3. SACMM方法。是一种软件架构修改的度量方法。



4. SASAM方法。通过对预期架构(架构设计阶段的相关描述材料)和实际架构(源代码中执行的架构)进行映射和比较来静态地评估软件架构。



5. ALRRA方法。是一种软件架构可靠性风险评估方法，该方法使用动态复杂度准则和动态耦合度准则来定义组件和连接件的复杂性因素，其中，动态复杂度准则在某个场景的执行中分析组件的动态行为来度量组件的复杂性，动态耦合度准则在某个场景的执行中分析连接件的消息传递协议来度量连接件的复杂性。利用失效模式和影响分析(FMEA)技术。



6. AHP(层次分析法)方法。是对定性问题进行定量分析的一种简便、灵活而又实用的多准则决策方法。AHP方法的特点是把复杂问题中的各种因素通过划分为相联系的有序层次使之条理化，并在一般情况下通过两两对比，根据一定客观现实的主观判断结构，把专家意见和分析者的客观判断结果直接、有效地结合起来，将一定层次上元素的某些重要性进行定量描述，之后利用数学方法计算反映每一层次元素的相对重要性次序的权值，并最后通过所有层次之间的总排序计算所有元素的相对权重及对权重进行排序。



7. COSMIC+UML方法。基于度量模型来评估软件架构可维护性的方法。针对不同表达方式的软件架构，采用统一的软件度量COSMIC方法来进行度量和评估。该方法主要是为了辅助分析软件架构的演化方案是否可行，并在开源软件DCMMS的软件架构UML组件图上得以验证。



考试真题



架构权衡分析方法(Architecture Tradeoff Analysis Method,ATAM)是一种系统架构评估方法, 主要在系统开发之前, 针对性能、(57)、安全性和可修改性等质量属性进行评价和折中。ATAM可以分为4个主要的活动阶段, 包括需求收集、(58)描述、属性模型构造和分析、架构决策与折中, 整个评估过程强调以(59)作为架构评估的核心概念。某软件公司采用ATAM进行软件架构评估, 在评估过程中识别出了多个关于质量属性的描述。其中, "系统在进行文件保存操作时, 应该与Windows系统的操作方式保持一致, 主要与(60)质量属性相关: "系统应该提供一个开放的API接口, 支持远程对系统的行为进行控制与调试, 主要与(61)质量属性相关。在识别出上述描述后, 通常采用(62)对质量属性的描述进行刻画与排序。在评估过程中, (63)是一个会影响多个质量属性的架构设计决策

(57)A.可测试性

B.可移植性

C.可用性

D.易用性

(58)A.架构视图

B.架构排序

C.架构风格

D.架构策略

(59)A.用例

B.视图

C.属性

D.模型

(60)A.可测试性

B.互操作性

C.可移植性

D.易用性

(61)A.可测试性

B.互操作性

C.可移植性

D.易用性

(62)A.期望管理矩阵

B.决策表

C.优先队列

D.效用树

(63)A.风险点

B.决策点

C.权衡点

D.敏感点



考试真题



【2022】58.改变加密级别可能会对安全性和性能产生非常重要的影响，因此，在软件架构评估中，该设计决策是一个()。

A.敏感点

B.风险点

C.权衡点

D.非风险点



【2022】59.效用树是采用架构权衡分析方法(Architecture Tradeoff Analysis Method,ATAM)进行架构评估的工具之一，其树形结构从根部到叶子节点依次为()。

A.树根、属性分类、优先级、质量属性场景

B.树根、质量属性、属性分类、质量属性场景

C.树根、优先级、质量属性、质量属性场景

D.树根、质量属性、属性分类、优先级



考试真题



【2022】63.在进行架构评估时，首先要明确具体的质量目标，并以之作为判定该架构优劣的标准。为得出这些目标而采用的机制叫做场景，场景是从()的角度对与系统的交互的简短描述。

A.用户

B.系统架构师

C.项目管理者

D.风险承担者

T H E E N D

功不唐捐，玉汝于成！

• 开 启 新 征 程 •

