




软件体系结构评估

- 
- 1. 软件体系结构评估概述
 - 2. ATAM评估方法
 - 3. SAAM评估方法

1. 软件体系结构评估概述

(1) 为什么要进行软件体系结构评估？

- 软件体系结构的好坏关系到软件产品的好坏，软件产品的好坏关系到软件公司的发展。
- 通过评估能了解系统的体系结构和重要属性(质量属性)，能够屏蔽风险，带来诸多收益。
- 到目前为止没有很好的自动化评估系统。

1. 软件体系结构评估概述

(2) 质量属性:

1) 可修改性

度量软件系统变化的成本。变化包括功能扩展、容量扩展、结构更新等。

2) 可用性

是指软件能够正常运行的时间比例。

可用性=平均工作时间/(平均工作时间+平均修复时间)

3) 性能

性能表征软件系统的响应速度或者由响应速度决定的其它度量。

1. 软件体系结构评估概述

(2) 质量属性:

4) 可测试性

可测试性表明软件系统在多大程度上容易被测试检查出缺陷。

5) 易用性

易用性表明软件系统完成后用户的体验和效率。

6) 安全性

安全性代表软件对未授权和非法操作的防卫能力。

1. 软件体系结构评估概述

(3) 评估的时机

- 前期
 - 又叫发现型评估
 - 找出难以实现的需求，并划分其优先级
- 后期
 - 为新系统或扩展评估老系统
 - 使用户明确是否可以通过更改老系统来满足新的质量以及功能需求

1. 软件体系结构评估概述

(4) 评估的必要性:

- 体系结构评估可以减少后期的测试和纠错的开销
- 评估是挖掘隐性需求并将其补充到设计的最后机会
- 体系结构是开发过程的中心，不良的体系结构会带来一塌糊涂的效果。

1. 软件体系结构评估概述

- (5) 评估的主要方式
 - 基于调查问卷或检查表的评估方式
 - 基于场景的评估方式
 - 基于度量的评估方式

1. 软件体系结构评估概述

- (5) 评估的主要方式
- 1) 基于调查问卷或检查表的评估方式
 - 调查问卷是一系列可以应用到各种体系结构评估的相关问题，其中有些问题可能涉及到体系结构的设计决策；有些问题涉及到体系结构的文档，有的问题针对体系结构描述本身的细节问题。
 - 检查表中也包含一系列比调查问卷更细节和具体的问题，它们更趋向于考察某些关心的质量属性。

1. 软件体系结构评估概述

- (5) 评估的主要方式
- 1) 基于调查问卷或检查表的评估方式
- 优点：
 - 这一评估方式比较自由灵活，可评估多种质量属性，也可以在软件体系结构设计的多个阶段进行。
- 缺点：
 - 由于评估的结果很大程度上来自评估人员的主观推断，因此不同的评估人员可能会产生不同甚至截然相反的结果，而且评估人员对领域的熟悉程度、是否具有丰富的相关经验也成为评估结果是否正确的重要因素。

1. 软件体系结构评估概述

- (5) 评估的主要方式
- 1) 基于调查问卷或检查表的评估方式
 - 尽管基于调查问卷与检查表的评估方式相对比较主观，但由于系统相关的人员的经验和知识是评估软件体系结构的重要信息来源，因而它仍然是进行软件体系结构评估的重要途径之一。

1. 软件体系结构评估概述

- (5) 评估的主要方式
- 2) 基于场景的评估方式
- 基于场景的方式由SEI首先提出并应用在体系结构权衡分析方法（ATAM）和软件体系结构分析方法（SAAM）中。
- 这种软件体系结构评估方式分析软件体系结构对场景也就是对系统的使用或修改活动的支持程度，从而判断该体系结构对这一场景所代表的质量需求的满足程度。
 - 例如，用一系列对软件的修改来反映易修改性方面的需求，用一系列攻击性操作来代表安全性方面的需求等。

1. 软件体系结构评估概述

- (5) 评估的主要方式
- 2) 基于场景的评估方式
- 优点：
 - 这一评估方式考虑到了包括系统的开发人员、维护人员、最终用户、管理人员、测试人员等在内的所有与系统相关的人员对质量的要求。
 - 基于场景的评估方式涉及到的基本活动包括确定应用领域的功能和软件体系结构的结构之间的映射，设计用于体现待评估质量属性的场景以及分析软件体系结构对场景的支持程度。

1. 软件体系结构评估概述

- (5) 评估的主要方式
- 2) 基于场景的评估方式
- 缺点：

- 不同的应用系统对同一质量属性的理解可能不同，例如，对操作系统来说，可移植性被理解为系统可在不同的硬件平台上运行，而对于普通的应用系统而言，可移植性往往是指该系统可在不同的操作系统上运行。由于存在这种不一致性，对一个领域适合的场景设计在另一个领域内未必合适，因此基于场景的评估方式是特定于领域的。
- 这一评估方式的实施者一方面需要有丰富的领域知识以对某以质量需求设计出合理的场景，另一方面，必须对待评估的软件体系结构有一定的了解以准确判断它是否支持场景描述的一系列活动。

1. 软件体系结构评估概述

- (5) 评估的主要方式
- 3) 基于度量的评估方式
 - 度量是指为软件产品的某一属性所赋予的数值，如代码行数、方法调用层数、构件个数等。
 - 传统的度量研究主要针对代码，但近年来也出现了一些针对高层设计的度量，软件体系结构度量即是其中之一。
 - 代码度量和代码质量之间存在着重要的联系，类似地，软件体系结构度量应该也能够作为评判质量的重要依据。

1. 软件体系结构评估概述

- (5) 评估的主要方式
- 3) 基于度量的评估方式
- 具体方法：
 - 赫尔辛基大学提出的基于模式挖掘的面向对象软件体系结构度量技术
 - Karlskrona和Ronneby提出的基于面向对象度量的软件体系结构可维护性评估
 - 西弗吉尼亚大学提出的软件体系结构度量方法等

1. 软件体系结构评估概述

(5) 评估的主要方式

4) 三种评估方式对比

	评估方式	调查问卷或检查表		场景	度量
		调查问卷	检查表		
通用性		通用	特定领域	特定系统	通用或特定领域
评估者对体系结构的了解程度		粗略了解	无限制	中等了解	精确了解
实施阶段		早	中	中	中
客观性		主观	主观	较主观	较客观

2. ATAM评估方法

- ATAM: Architecture Tradeoff Analysis Method
(体系结构权衡分析方法)
- (1) 目标
 - 理解SA关于软件系统的质量属性需求决策的结果。
 - 这些设计决策会一直影响软件生命周期，在软件实现后很难改变这些决策。
- (2) 特点
 - 揭示SA对特定质量目标的满足情况，描述诸多质量目标之间的相互作用和权衡
 - 结构化的评估方法，可重复

2. ATAM评估方法

- (3) 基本概念
 - 敏感点 (Sensitivity point)
 - 敏感点是一个或多个构件的特征
 - 敏感点可以使设计师搞清楚实现质量目标时应该注意什么
 - 权衡点 (Tradeoff point)
 - 权衡点是影响多个质量属性的特征
 - 是多个质量属性的敏感点
 - 权衡点需要进行权衡

2. ATAM评估方法

- (3) 基本概念
 - 敏感点影响一个质量属性
 - 权衡点影响多个质量属性
- Eg. 2010年软件架构师考试题
 - ____ (62) ____是实现一个特定质量属性的关键特征，该特征为一个或多个软件构件所共有。
 - “改变加密的级别可能会对安全性和性能都产生显著的影响”，这是一个对系统____ (63) ____的描述。

2. ATAM评估方法

- (3) 基本概念
 - 风险承担者、涉众、牵涉到的人 (Stakeholders)
 - 体系结构设计师
 - 开发人员
 - 维护人员
 - 集成人员
 - 测试人员
 - 标准专家
 - 性能工程师

2. ATAM评估方法

- (3) 基本概念

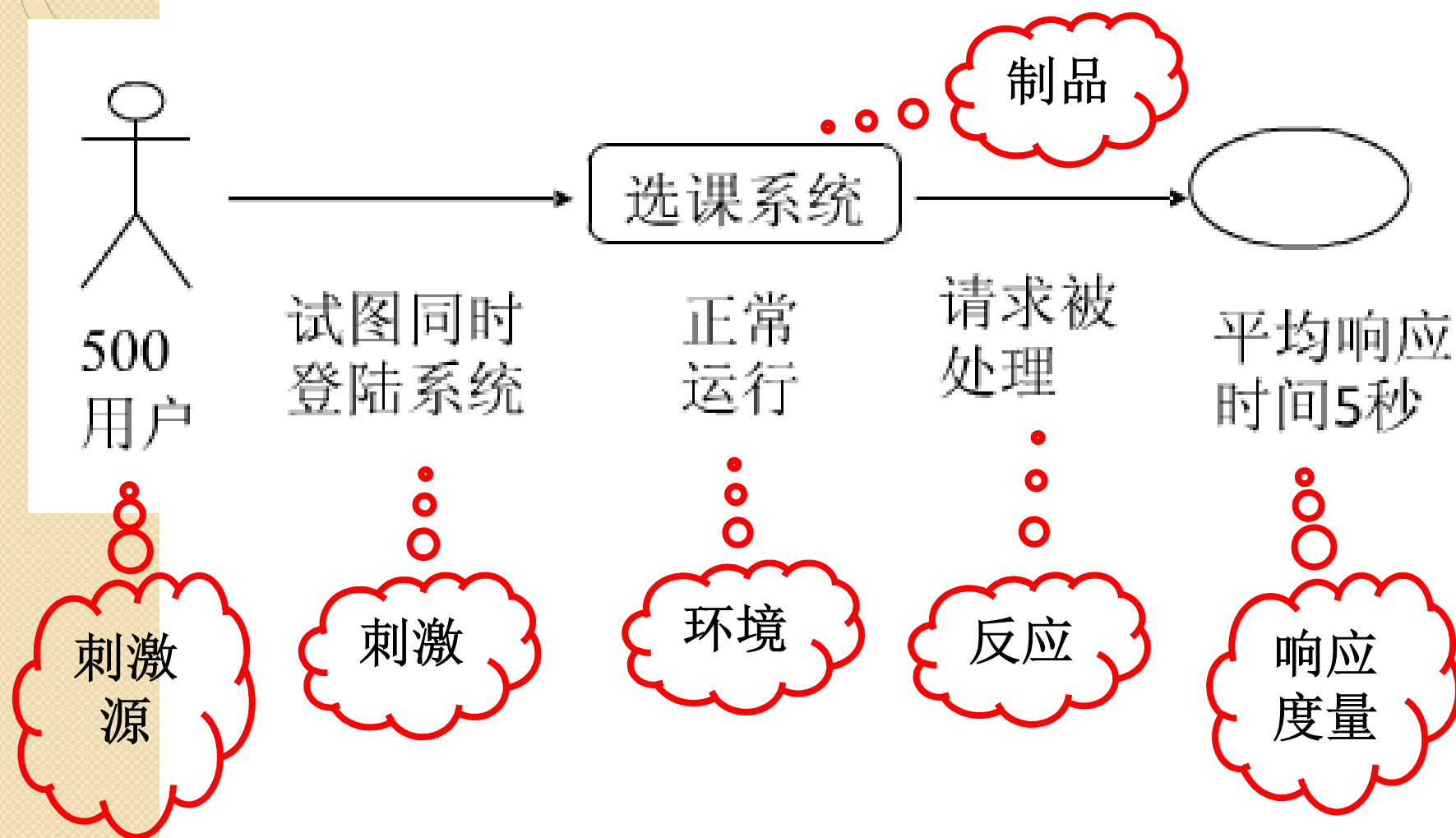
- 场景

- 在进行体系结构评估时，一般首先要精确地得出具体的质量目标，并以之作为判定该体系结构优劣的标准。我们把为得出这些目标而采用的机制叫做场景。
- 场景是从风险承担者的角度对与系统的交互的简短描述。
- 在体系结构评估中，一般采用刺激、环境和响应三方面来对场景进行描述

2. ATAM评估方法

- **刺激**是场景中解释或描述风险承担者怎样引发与系统的交互部分。例如，用户可能会激发某个功能，维护人员可能会做某个更改，测试人员可能会执行某种测试等，这些都属于对场景的刺激。
- **环境**描述的是刺激发生时的情况。例如，当前系统处于什么状态？有什么特殊的约束条件？系统的负载是否很大？某个网络通道是否出现了阻塞等。
- **响应**是指系统是如何通过体系结构对刺激作出反应的。例如，用户所要求的功能是否得到满足？维护人员的修改是否成功？测试人员的测试是否成功等。

2. ATAM评估方法



2. ATAM评估方法

（4）ATAM的过程

- 一个完整的ATAM通常会花费三个整天，其中每天都要进行如下三个过程：场景提出、架构提出和将场景映射到架构上，并进行分析评估。

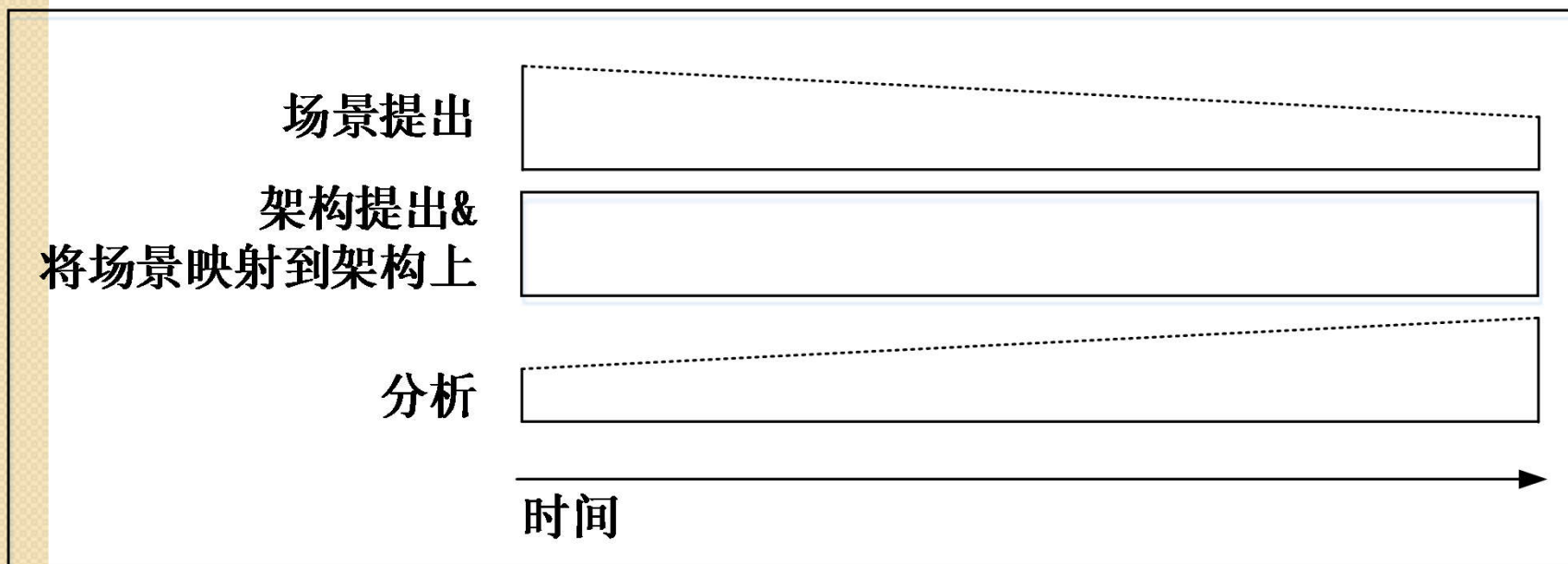


图7-11 ATAM活动及其重要性对比图

2. ATAM评估方法

(5) ATAM的步骤

ATAM主要部分包括4组，共9个步骤：

1) 陈述，包括通过它进行的信息交流

- ① ATAM方法的陈述：评估负责人
- ② 商业动机的陈述：项目经理或系统客户
- ③ SA的陈述：系统设计人员

2. ATAM评估方法

(5) ATAM的步骤

2) 调查与分析，包括对照体系结构方法评估关键质量属性需求

- ④ 确定体系结构方法：系统设计人员
- ⑤ 生成质量属性效用树（utility tree）：说明构成系统“效用”的质量属性（性能、有效性、安全性、可修改性、可用性），具体到场景层次，标注刺激/反应，并区分不同的优先级
- ⑥ 分析体系结构方法：基于步骤5识别出的高优先级的场景，说明和分析针对这些场景的体系结构方法。在这一步骤中，体系结构风险、非风险、敏感点和权衡点被识别出来

2. ATAM评估方法

(5) ATAM的步骤

3) 测试，包括对照所有相关人员的需求检验最新结果

- ⑦ 集体讨论并确定场景优先级
- ⑧ 分析体系结构方法：针对步骤7的高等级场景

4) 形成报告，包括陈述ATAM的结果

- ⑨ 结果的表述：包括方法、场景、特定属性的问题、效用树、有风险决策、无风险决策、敏感点和权衡点

提示：上述步骤顺序并不严格，可根据需要适当调整

2. ATAM评估方法

- (6) ATAM的步骤解析
- 第1步：ATAM方法的陈述
 - 评估负责人向参加会议的相关人员介绍ATAM方法。在这一步，要对每个人解释参与的过程，使每个人都知道要收集哪些信息，如何描述信息，将向谁报告等等。
 - 进行ATAM评估步骤简介
 - 介绍用于获取信息和分析的技巧：效用树的生成、基于体系结构方法的获取和分析、对场景的集体讨论及优先级的划分
 - 描述评估的结果：场景及其优先级、用户理解和评估体系结构的问题、描述体系结构的动机需求并给出其优先级的效用树、所确定的一组体系结构方法、所发现的有风险决策、无风险决策、敏感点和权衡点等

2. ATAM评估方法

- (6) ATAM的步骤解析
- 第2步：业务动机的陈述
 - 项目经理从业务角度，向相关人员介绍系统概况
 - 系统最重要的功能
 - 技术、管理、经济、政治方面的任何相关限制
 - 业务目标和上下文
 - 主要的相关人员
 - 体系结构的驱动因素，即促使形成该体系结构的主要质量属性目标

2. ATAM评估方法

- (6) ATAM的步骤解析
- 第3步：体系结构陈述
 - 架构师在适合的细节层次上描述体系结构。这些体系结构信息直接影响可能的分析及分析的质量。在进行更实质的分析之前，评估小组通常需要询问更多的有关体系结构的信息，包括：
 - 技术约束条件，诸如要求使用的操作系统、硬件、中间件等
 - 该系统必须要与之交互的其他系统
 - 用以满足质量属性需求的体系结构方法、风格、模式和采用的机制
 - 高层体系结构视图：功能、代码、并发、物理

2. ATAM评估方法

- (6) ATAM的步骤解析
- 第4步：确定体系结构方法
 - 体系结构方法定义了系统的重要结构，描述了系统演化、对更改的响应、对攻击的防范以及与其他系统的集成等
 - 强调体系结构方法和体系结构风格的确定，代表了所评估的体系结构用以事先具有高优先级的质量属性的手段，保证关键需求按计划得以实现的手段
 - 体系结构风格，包括构件类型及其拓扑结构的描述，对构件间数据和控制交互模式的描述和使用该样风格的优缺点的非正式表述
 - 体系结构的约束条件，对构件及其交互的约束，这些约束条件限定了满足质量属性需求的设计决策

2. ATAM评估方法

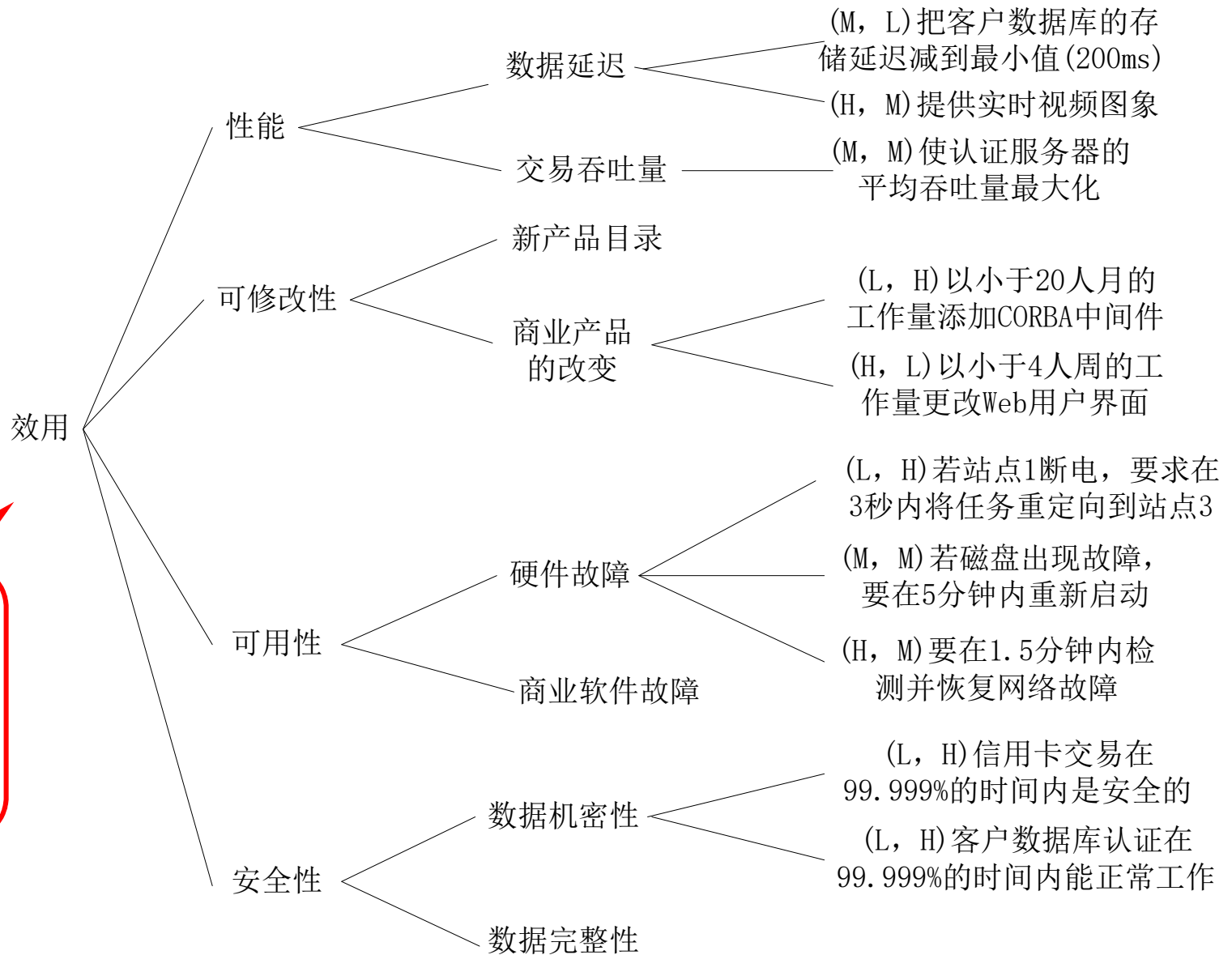
- (6) ATAM的步骤解析
- 第5步：生成质量属性效用树
 - 评估小组与项目决策者合作，共同确定出该系统的最重要的质量属性目标，并设置优先级，进行进一步的细化，该步指导其他的分析
 - 这种方式将所有风险承担者和评估小组的精力集中到对系统的成功与否具有重要意义的体系结构的方面上

效用树为我们提供了一种直接而有效地将系统的业务驱动因素转换为具体的质量属性场景的机制，该步骤的输出结果是对具体质量属性需求（以场景形式实现）的优先级的确定。

- 效用树中质量属性细化为场景
- 确定最重要的质量属性目标，并设置优先级
- 效用树设置优先级标准
 - 每个场景对系统成功与否的重要性
 - 体系结构设计人员所估计的实现这种场景的难度

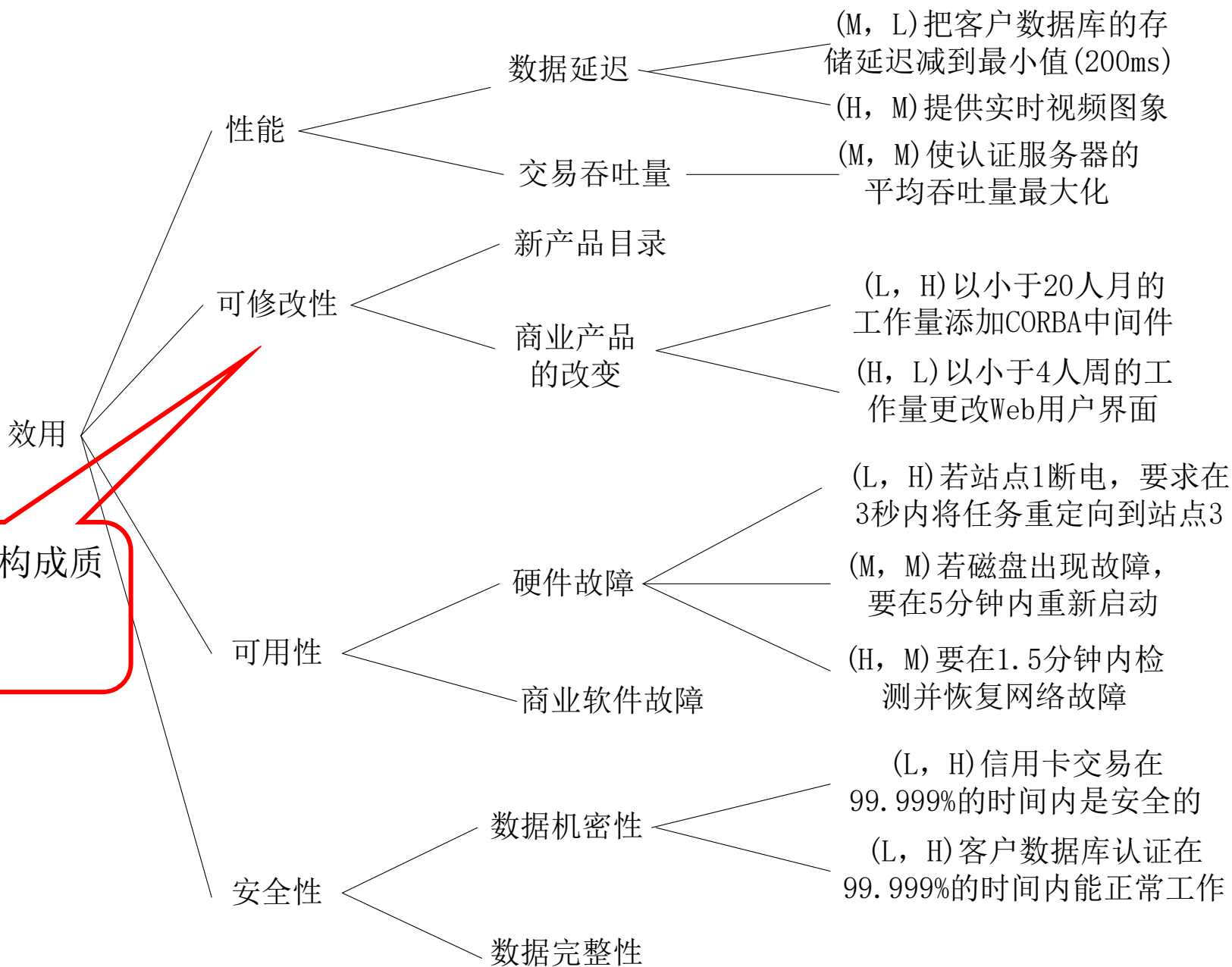
效用树样例

代表系统的
整体质量

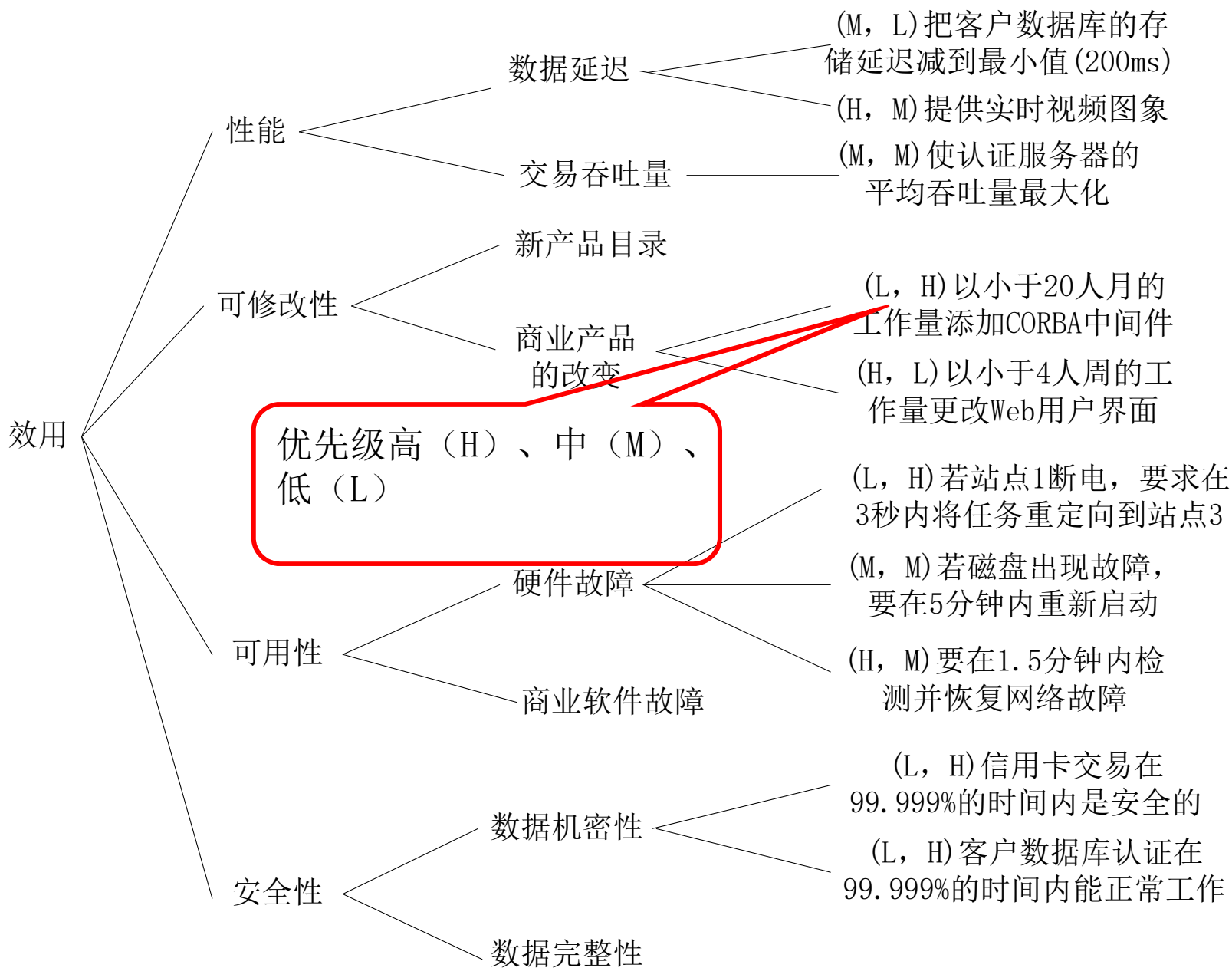


效用树样例

二级节点构成质量属性



效用树样例



2. ATAM评估方法

- (6) ATAM的步骤解析
- 第6步：分析体系结构方法
 - 针对划分了优先级的质量需求（第5步）和采用的体系结构方法（第4步），评估它们的匹配情况
 - 把效用树中每个高优先级的质量属性需求与实现它们的体系结构方法关联起来
 - 为每一个效用树生成的高级场景确定构件、连接件、配置和约束。
 - 体系结构设计师通过问一系列方法特定和质量属性特定的问题来实现每个体系结构方法。
 - 风险决策、无风险决策、敏感点、权衡点的确认

2. ATAM评估方法

* 体系结构方法分析模板

构架方法分析				
场景号:序号	场景:来自效用树的对场景的文字表述			
属性	该场景所关注的质量属性			
环境	对系统所依赖的环境的相关假设,以及场景执行时的相关条件			
刺激	对该场景所体现的质量属性刺激(如功能调用、故障、安全威胁、所做更改等)的精确叙述			
响应	对质量属性响应(如响应时间、所做更改难度的度量等)的精确叙述			
构架决策	敏感点	权衡点	有风险决策	无风险决策
与该场景相关的影响质量属性响应的构架决策	敏感点序号	权衡点序号	有风险决策序号	无风险决策序号
.....
.....
推理	定性的或量化的关于为什么这组构架决策能够满足此场景所表达的每个质量属性要求的基本原理			
构架图	一个或多个表示构架视图的图形,其中要标注出支持上述推理的构架信息,如有必要也可带有解释性文字叙述			

* 举例：
体系结构
方法分析
示例

构架方法分析				
场景号: A12	场景: 检测主 CPU 的硬件故障并使系统从中恢复			
属性	可用性			
环境	正常操作			
刺激	某个 CPU 不能正常工作了			
响应	切换可用概率是 0.999999			
构架决策	敏感点	权衡点	有风险决策	无风险决策
备用 CPU	S2		R8	
无备用数据通道	S3	T3	R9	
看门狗	S4			N12
心跳	S5			N13
故障切换路由	S6			N14
推理	<ul style="list-style-type: none">• 通过使用不同的硬件和操作系统保证不出现通用模式故障(参见有风险决策 R8)• 最坏情况下, 4 秒钟(即运算状态的最长时间)之内完成恢复。• 根据心跳和看门狗的速度, 保证在 2 秒钟之内检测到故障• 看门狗简单可靠(已经过验证)• 由于缺少备用数据通道, 因此可用性可能存在风险(参见有风险决策 R9)			
构架图				
<pre>graph LR; In(()) --> MainCPU[主CPU (操作系统1)]; In --> BackupCPU[备用CPU 和看门狗 (操作系统2)]; MainCPU -- "心跳 (1秒)" --> BackupCPU; MainCPU --> SwitchCPU[切换CPU (操作系统1)]; BackupCPU --> SwitchCPU; SwitchCPU --> Out(())</pre>				

- (6) ATAM的步骤解析
- 第7步：集体讨论并确定场景优先级
 - 在第7步和第8步，评估组测试所理解的体系结构，场景被用作测试的主要手段。第5步确定的场景主要是从体系结构设计人员的角度看待系统的质量属性需求，这一步是从相关人员的角度讨论场景
 - 需讨论的场景：用例场景、成长场景、考察场景，这些场景可能同效用树的场景保持一致，也可能发现更多的驱动场景
 - 用例场景：描述风险承担者期望使用系统的方式。
 - 成长场景：体系结构在中短期的改变，包括功能和非功能的修改。成长场景能够使评估人员看清在预期因素影响系统时，体系结构表现出来的优缺点。
 - 考察场景：描述系统成长的极端情形。例如：性能中数量级的改变，任务的重大变更等。考察场景试图找出敏感点和权衡点，有助于评估系统质量属性的限制。

2. ATAM评估方法

- 第7步的具体步骤：
 - 场景收集；
 - 相关人员对场景投票确定优先级；
 - 比较场景讨论结果和质量效用树，找出其中相同之处和不同之处；
 - 在高优先级的场景和效用树中的高优先级的结点之间的任何差异都要重新调整，至少要有合理解释；
 - 将场景讨论结果放到质量效用树当中，即系统体系结构设计和系统需求一致

- 集体讨论并确定场景优先级
 - 一旦收集了若干个场景后，必须要设置优先级。评估人员可通过投票表决的方式来完成，每个风险承担者分配相当于总场景数的30%的选票，且此数值只入不舍。例如，如果共有17个场景，则每个风险承担者将拿到6张选票，这6张选票的具体使用则取决于风险承担者，他可以把这6张票全部投给某一个场景，或者每个场景投2-3张票，还可以一个场景一张票等。
 - 一旦投票结果确定，所有场景就可设置优先级。设置优先级和投票的过程既可公开也可保密。

- 对某车辆调度系统进行评估时所得到的几个场景及其得票情况

场景编号	场景描述	得票数量
4	在 10 分钟内动态地对某次任务得重新安排	28
27	把对一组车辆得管理分配给多个控制站点	26
10	在不重新启动系统的情况下，改变已开始任务的分析工具	23
12	在发出指令后 10 秒内，完成对不同车辆的重新分配，以处理紧急情况	13
14	在 6 人月内将数据分配机制从 CORBA 改变为新兴的标准	12

[Next](#)

- 场景与质量属性

场景编号	得票数量	质量属性
4	28	性能
27	26	性能、可修改性、可用性
10	23	可修改性
12	13	性能
14	12	可修改性

- 效用树生成和场景集体讨论的差异

	效用树	场景集体讨论
风险承担者	构架设计师、项目负责人	所有风险承担者
一般的人员规模	评估人员;2~3 位项目人员	评估人员;5~10 位项目人员
主要目标	得出促成该构架的主要质量属性,使这些属性更为具体,并为其设置优先级为以后的评估确定所关注的焦点	促进风险承担者之间的交流,验证通过效用树所得出的质量属性目标
方法	从一般到具体:从质量属性开始,不断求精,直至得到场景	从具体到一般:从场景开始,最终确定出它们所表达的质量属性

2. ATAM评估方法

- (6) ATAM的步骤解析
- 第8步：分析体系结构方法
 - 在已确定了若干场景并进行了分析之后，评估小组就可以引导体系结构设计师在所描述的体系结构的基础上实现第7步中得出的最高优先级的场景，对相关的体系结构决策如何有助于该场景的实现做出解释
 - 与第6步类似
 - 对新增的场景，分析其体系结构方法；对不变的场景，进行检查

2. ATAM评估方法

- (6) ATAM的步骤解析
- 第9步：陈述结果
 - 最后，需要把在ATAM分析中所得到的各种信息进行归纳总结，并呈现给相关人员
 - 在这一陈述中，评估负责人概要介绍ATAM评估的各个步骤和得到的各种信息，包括商业环境、促成该体系结构的主要需求、约束条件和体系结构等，但最重要的结果如下：
 - 文档化的体系结构方法
 - 若干场景及其优先级
 - 基于质量属性的若干问题
 - 效用树
 - 风险决策、无风险决策、敏感点、权衡点

2. ATAM评估方法

- (7) ATAM方法总结
- 以时间为维度，评估分为4个阶段，
 - 第0阶段：建立阶段
 - 合作关系的建立——评估客户和评估人员之间
 - 准备工作——组建评估小组，召开开工会议，进行准备
 - 第1阶段：第1~6步
 - 以体系结构为中心
 - 重点是获取体系结构信息并对其进行分析
 - 第2阶段：第7~9步
 - 以相关人员为中心
 - 重点是获知相关人员的观点，并验证第1阶段的结果
 - 第3阶段：后续阶段
 - 形成最终报告、对后续活动（如果有的话）做出规划
 - 评估小组在此阶段实现文档和经验的更新

2. ATAM评估方法

- (7) ATAM方法总结
- 第0阶段：
- 合作关系的建立
 - 评估人员和评估客户的交流
 - 评估客户应该是对所评估体系结构对应的项目有一定影响力的人，而且可以联系到很多位体系结构相关人员
 - 客户应对所要采用的评估方法有基本了解，并且知道在评估过程中都要做哪些工作
 - 客户应对所要评估的体系结构及其系统做出描述
 - 假设评估负责人已经决定可以进行评估，则应商谈并签署关于评估工作的合同或协议
 - 要解决好信息专有性问题。例如，评估小组可能需要签署不得泄漏该评估信息的协议

2. ATAM评估方法

- (7) ATAM方法总结
- 第0阶段:
- 准备工作
 - 组建评估小组
 - 角色: 评估小组负责人, 评估负责人, 场景书记员, 进展书记员, 计时员, 过程观察员, 过程监督者, 提问者
 - 召开评估小组开工会议
 - 就评估的经验和体会进行广泛的交流, 同时指定每个成员要扮演的角色
 - 为第1阶段进行必要的准备

2. ATAM评估方法

- (7) ATAM方法总结
- 第一阶段:
- 完成第1~6步的工作，通过收集足够多的信息，以决定：
 - 之后的评估工作是否可行、能否顺利展开
 - 如果不行，就可在为第2阶段的工作召集更多的相关人员之前，在第1阶段及时终止
 - 是否需要更多的体系结构文档
 - 如果需要，则应明确需要的是哪些类型的文档，以及应以什么形式提交这些文档
 - 哪些相关人员应参与第2阶段的工作
 - 在第一阶段的最后，评估出资人要保证让合适的相关人员参与第2阶段的工作

2. ATAM评估方法

(7) ATAM方法总结

第二阶段：在简要重复第1~6步工作的基础上，进行第7~9步的工作

步骤	所做工作	第1阶段的参与者	第2阶段的参与者
1	ATAM方法的陈述	评估小组和项目决策者	评估小组、项目决策者和所有相关人员
2	商业动机的陈述		
3	体系结构的陈述		
4	确定体系结构方法		
5	生成质量属性效用树		
6	分析体系结构方法		
7	集体讨论并确定场景优先级	——	
8	分析体系结构方法		
9	评估结果陈述		

2. ATAM评估方法

- (7) ATAM方法总结
- 第三阶段：
- 生成最终报告
 - 在ATAM评估快要结束时，必须撰写并提交最终的评估报告：做了哪些工作、有何发现、得出了什么结论等
- 收集数据
 - 针对参与者/评估小组成员的改进意见调查，询问对评估实践的看法
 - 针对评估客户/评估小组成员的成本调查
 - 针对评估客户的长期收益调查
- 更新工作产品仓库
 - 对在刚完成的评估中所用到的或所生成的工作产品进行维护，这将有助于更好地进行未来的评估工作，包括：成本/收益信息、场景、待分析的问题、参与者的意见、评估的最终报告

2. ATAM评估方法

针对第1阶段	针对第2阶段
较少的相关人员(2~4人)	许多相关人员(10~15人)
以体系结构为中心	以相关人员为中心
面向解决方案	面向问题
面向分析	面向验证
促进对体系结构的理解	促进相关人员之间的交流
场景主要用于效用树的创建	场景主要用于效用树的验证
大信息量的、非正式的技术上的交流	有组织的会议

2. ATAM评估方法

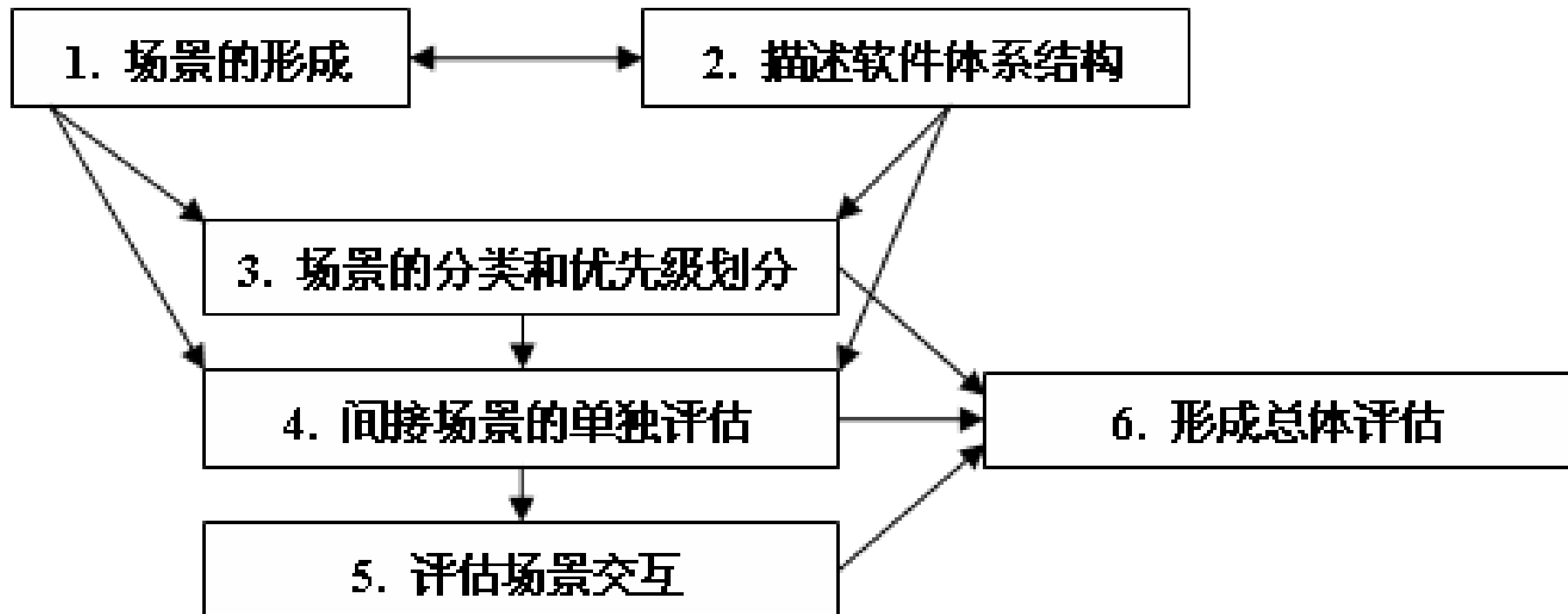
- (8) ATAM方法案例
- 票务系统架构评审案例分析

3. SAAM评估方法

- SAAM (Software Architecture Analysis Method)
主要分析体系结构的可修改性；也可以对系统属性及系统功能进行分析
- 评估方法
 - 描述场景、划分优先级、分析体系结构问题
- SAAM评估的输入
 - 一组场景
- SAAM评估的输出
 - 将代表了未来可能做的更改的场景与体系结构对应起来，显示出体系结构中未来可能具有较高复杂性的地方，估计预期工作量
 - 理解系统功能，比较多个体系结构支持的功能数量

3. SAAM评估方法

* SAAM评估步骤



3. SAAM评估方法

- SAAM评估步骤
- 第1步：场景的形成
 - 集体讨论形成场景
 - 全面捕捉系统的主要用途、系统用户类型、系统将来可能的变更、系统在当前及可预见未来必须满足的质量属性
 - 场景的提出和收集过程可以重复执行
- 第2步：体系结构的表述
 - 描述系统的静态特征和动态特征
 - 场景的形成和体系结构的表述相互促进，反复迭代。

3. SAAM评估方法

- 第3步：场景的分类和优先级的确定
 - 场景分类：直接场景 / 间接场景
 - 直接场景：这一预计的使用情况不需要对体系结构做任何修改就可以实现
 - 间接场景：所评估的体系结构不能直接支持某一场景，必须对该体系结构进行一定的修改才能实现的场景。
 - 划分优先级，考察重要场景（投票选出）
- 第4步：对场景的单个评估
 - 场景与体系结构描述对应起来
 - 对直接场景，说明体系结构如何执行
 - 对间接场景，说明更改及其代价

3. SAAM评估方法

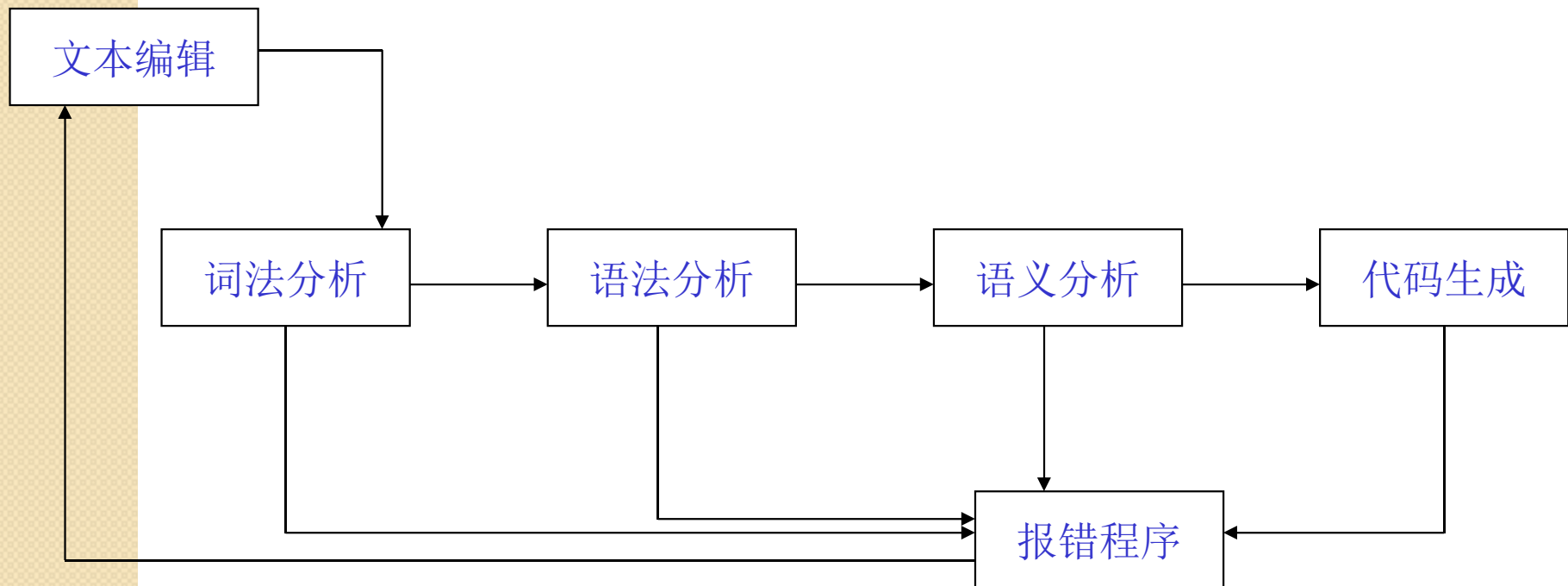
- 第5步：场景相互作用的评估
 - 场景相互作用：两个或者多个间接场景要求更改同一个构件
 - 体现体系结构设计中的功能分配
 - 暴露体系结构文档未充分说明结构的分解
- 第6步：形成总体评估
 - 为场景设置权值
 - 体系结构支持的直接场景数量

实例：CASL编译程序软件体系结构分析

- CASL语言的编译系统逻辑上分为6个功能：
 - (1) 文本编辑器：一个类似与写字板的文本编辑器，用户可用来输入程序代码；还用来显示编译系统的界面元素
 - (2) 词法分析器：对用户输入的程序代码进行词法分析。若有错，则进入报错程序。
 - (3) 语法分析器：对用户输入的程序代码进行语法分析。若有错, 则进入报错程序。
 - (4) 语义分析器：对用户输入的程序代码进行语义分析。若有错，则进入报错程序。
 - (5) 代码生成器：以上步骤若没有错误则生成可执行代码。
 - (6) 报错程序：报告错误的行数。

3. SAAM评估方法

* CASL语言编译系统体系结构



3. SAAM评估方法

- (1) 其他格式的文档，比如以前只支持本软件自定义的文档格式，不支持普通的文本 (*.txt) 格式和 Word 文档 (*.doc) 格式，现在要求支持普通文本格式和 Word 文档格式等。
- (2) 增加文档结构索引的显示风格。
- (3) 格式长度发生改变，比如指令格式长度从原来的16位升为要求的32位。
- (4) 移植到其他的操作系统，比如UNIX，LINUX操作系统等。
- (5) 从对一个程序文本的编译改为对一个程序工程的编译。
- (6) 与其他的软件结合。

- (1) 支持其他不同格式的文档：间接支持，需要改动的模块是文本编辑模块。
- (2) 改变文档结构索引表的显示风格：间接支持，需要改动的模块是文本编辑模块。
- (3) 改变指令格式长度：间接支持，需要改动的模块是代码生成模块。
- (4) 移植到其他的操作系统之中：间接支持，所有调用WinAPI函数的模块都需要改动，包括：文本编辑模块，词法分析模块，语法分析模块，语义分析模块，代码生成模块，报错程序模块等。
- (5) 从对一个程序文本的编译改为对一个程序工程的编译：间接支持，需要改动的模块是代码生成模块。
- (6) 与一个新的开发工具结合：间接支持，需要把CASL语言编译系统改为可嵌入的服务器软件，需要改动的模块是文本编辑模块。

模块	改动数目
文本编辑	4
词法分析	1
语法分析	1
语义分析	1
代码生成	3
报错程序	1

3. SAAM评估方法

- 应用SAAM方法对CASL语言编译系统体系结构进行分析后,可以得出如下结论:
- (1) 允许设计者将整个系统的输入/输出行为看成是多个过滤器的行为的简单合成。
- (2) 系统的各个模块抽象得比较好,具有良好的隐蔽性和高内聚、低耦合的特点。
- (3) 数据格式的改变对体系结构没有很大的影响。
- (4) 由于采用了面向对象的体系结构和基于管道过滤器的体系结构,所以各个模块的可重用性比较强,系统维护和增强系统性能简单,新的过滤器可以添加到现有系统中来;旧的可以被改进的过滤器替换掉。

3. SAAM评估方法

总之，CASL语言编译系统的体系结构具有较好的特性，能够很好地满足用户的需求。但是也存在不足之处，如涉及到用户界面的模块只有1个，这说明没有对功能模块进行细化，面临诸多问题，如：可移植性、可修改性等方面的问题时都要改动这个模块，这是需要改进的地方。