

软件体系结构

SEI/BUAA

北航计算机学院

School of Computer Science & Engineering

第三部分

创建软件体系结构

张莉 教授

北京航空航天大学

软件工程研究所

lily@buaa.edu.cn

软件体系结构

SEI/BUAA

创建软件体系结构

第三部分

创建软件体系结构

理解 and 实现质量属性

- 软件系统的质量属性
- 软件体系结构与质量属性
- 软件体系结构设计的原子操作
- 软件体系结构战术
- 软件体系结构模式和风格

软件体系结构案例分析

- KWIC System
- WWW
- CORBA

北京航空航天大学软件工程研究所

软件体系结构

SEI/BUAA

创建软件体系结构

理解质量属性

“Cheshire-Puss,”[Alice] began, rather timidly...”Would you tell me, please, which way I ought to go from here?”

“That depends a good deal on where you want to go”,said the Cat.

“Oh, I don’t much care where—”said Alice.

“Then it doesn’t matter which way you go,”said said the Cat

“--so long as I get somewhere,” said Alice.

“Oh, you’re sure to do that”, said the cat, “If only you walk long enough”

商业目的决定了系统体系结构必须满足的一些质量属性

这些质量属性是系统功能（系统能力、服务和行为）的基本要求之上的

北京航空航天大学软件工程研究所

当前，对软件体系结构设计的共识之一是，体系结构设计应当支持对软件系统质量的需求。

Before --- Quality Attributes

软件体系结构

SEI/BUAA

创建软件体系结构

Architecting a dog house

北京航空航天大学软件工程研究所

软件体系结构

SEI/BUAA

创建软件体系结构

Architecting a house

北京航空航天大学软件工程研究所

软件体系结构

Architecting a high rise

创建软件体系结构

北京航空航天大学软件工程研究所

软件体系结构

Early architecture

创建软件体系结构

**Progress**  
 - Limited knowledge of theory

北京航空航天大学软件工程研究所

软件体系结构

Modern architecture

创建软件体系结构

**Progress**  
 - Advances in materials  
 - Advances in analysis  
  
**Scale**  
 - 5 times the span of the Pantheon  
 - 3 times the height of Cheops

北京航空航天大学软件工程研究所

软件体系结构

Modeling a house

创建软件体系结构

北京航空航天大学软件工程研究所

软件体系结构

Movements in civil architecture

体系结构

- Bronze age/Egyptian (Imhotep)
- Grecian/Roman (Vitruvius)
- Byzantine/Romanesque
- Gothic
- Mannerism (Michelangelo, Palladio)
- Baroque
- Engineering/Rational/National/Romantic
- Art nouveau
- Modern movement (Wright, LeCorbusier)

**Progress**  
 - Imitation of previous efforts  
 - Learning from failure  
 - Integration of other forces  
 - Experimentation

Architecture Style

北京航空航天大学软件工程研究所

软件体系结构

Kinds of civil architecture

软件体系结构

- **Community**
  - houses, flats and apartments, gardens, education, hospitals, religion
- **Commerce**
  - shops and stores, restaurants, hotels, office buildings, banks, airports
- **Industry**
  - industrial buildings, laboratories, farm buildings
- **Leisure**
  - sport, theatres

领域：参考体系结构

北京航空航天大学软件工程研究所

软件体系结构 SEI/BUAA 软件体系结构

## Forces in civil architecture

**Kinds of loads**

- Dead loads
- Live loads
- Dynamic loads

**Avoiding failure**

- Safety factors
- Redundancy
- Equilibrium

Any time you depart from established practice, make ten times the effort, ten times the investigation. Especially on a very large project.  
- LeMessurier

北京航空航天大学软件工程研究所

软件体系结构 SEI/BUAA 创建软件体系结构

## Forces in Software

北京航空航天大学软件工程研究所

软件体系结构 SEI/BUAA

## 第二部分 创建软件体系结构

- 理解和实现质量属性
  - 软件系统的质量属性
    - 软件体系结构与质量属性
    - 软件体系结构战术
    - 软件体系结构风格
    - 软件体系结构设计的原子操作
- 软件体系结构案例分析
  - KWIC System
  - WWW
  - CORBA

北京航空航天大学软件工程研究所

软件体系结构 SEI/BUAA 创建软件体系结构

## 软件体系结构和质量属性

- 功能性需求和非功能性需求
  - 功能性需求是在首位考虑的。
  - 很多情况下我们对系统进行重新设计不是由于功能上的不足，而是由于系统的维护、移植或扩展十分困难，系统运行速度太慢，或者是因为系统遭到了网络黑客的攻击。
  - 系统功能与软件结构的映射关系决定了软件体系结构对质量属性的支持度。这种是折中的：如果不牺牲系统的其他一个或多个质量属性，就无法最大限度的实现任何一个质量属性。
  - 非功能需求包括除了功能需求所指内容之外的所有质量属性，这个词本身不确切。

北京航空航天大学软件工程研究所

软件体系结构 SEI/BUAA 创建软件体系结构

## 功能性和体系结构

- 功能性和质量属性是正交的。
  - 关注点分离。
- 什么是功能性？
  - 指系统能够完成所希望的工作的能力。
- 和体系结构的关系
  - 功能需要多元素的协作完成。如果各元素的责任划分不合理，系统就不可能提供所需的功能；
  - 大量结构中，可以通过任何一个结构来实现功能。因此，功能在很大程度上是独立于结构的；
  - 功能性关心：它是如何与其他质量属性交互，以及它是如何限制其他质量属性的。

北京航空航天大学软件工程研究所

软件体系结构 SEI/BUAA 创建软件体系结构

## 体系结构和质量属性

- 必须从设计、实现到部署的整个过程中考虑质量属性。
  - 如：易用性涉及体系结构和非体系结构因素
  - 可修改性由功能划分的方式（体系结构方面的）和模块中的编码技巧（非体系结构方面的，如可读性）决定。
  - 系统性能既依赖于体系结构又不完全依赖于体系结构。
  - 结论：
    - 对于大多数质量属性，体系结构具有重要意义。再体系结构设计时应考虑，并评估。
    - 体系结构不能独立地实现质量属性，它提供了实现质量属性地基础

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

在复杂系统中，**决不可能以孤立的方式实现质量属性。**

- 任何一个属性的实现都会对其他质量属性的实现带来积极或消极的影响。
- 如：安全性和可靠性
- 几乎每个质量属性都会对系统的性能产生消极的影响，如可移植性。

北京航空航天大学软件工程研究所

软件体系结构

系统的质量属性

创建软件体系结构

Boehm模型利用分解的思想来描述软件质量

大学软件工程研究所

软件体系结构

创建软件体系结构

McCall模型对质量进行分解，并将质量属性按照产品的生命周期进行分类。

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

ISO/IEC9126是McCall模型的派生模型。将软件质量分解为6个特性：

- 功能性
- 可靠性
- 效率
- 可使用性
- 可维护性
- 可移植性

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

为了方便起见，从以下两个方面讨论质量属性：

- 通过运行可观察到的系统质量属性
  - 在运行过程中，系统满足其行为需求的程度如何？系统运行是否得到了所要求的结果？结果的得出是否及时？结果是否正确或者结果是否达到了规定的精度或稳定性要求？当与其他系统连接时，系统工作是否正常？
- 通过运行不可观察到的系统质量属性
  - 系统集成、测试和修改的难易程度如何？开发的成本有多高？是在什么时间投放市场的？

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

## 软件体系结构和质量属性

我们将从以下三个方面讨论:

- 系统的**质量属性**
  - 能通过观察系统运行确定的质量属性
  - 不能通过观察系统运行确定的质量属性
- 受体系结构影响的商业属性**
- 与软件体系结构本身相关的一些质量属性, 如概念完整性。**

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

## 直接归属系统的执行属性

- 可在运行时观察确定的系统质量属性**

质量属性	体系结构策略
<b>性能</b>	系统内各组件的交互; 功能划分以开发系统的并行性。
<b>安全性</b>	特殊组件, 如安全内核或认证服务器。
<b>可用性</b>	采用冗余组件增加容错性; 控制组件的交互。
<b>功能性</b>	与其他质量属性的交互。
<b>易用性</b>	易用性是一个积极因素; 获得正确的信息流; 效率与性能的关系

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

## 性能 (Performance)

[定义]: 对外部刺激 (事件) 做出反应所需要的时间或在某个时间内所处理的事件个数。

- 性能这一质量属性经常用在单位时间内所能完成的处理数量或系统为一个处理所要消耗的时间来表示。
- 由于通信所用的时间通常长于计算所用的时间, 因此系统的性能通过是系统中各个组件间进行通信或交互的数据量的函数——这显然是软件体系结构问题。

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

## 安全性 (Security)

[定义]: 是衡量系统在向合法用户正常提供服务的情况下, 阻止企图非授权使用和抗拒拒绝服务攻击的能力。

包括:

- 拒绝服务
- IP源地址欺骗

软件体系结构策略:

- 认证服务器
- 网络监视器
- 防火墙
- 可信的内核

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

## 可用性 (Availability)

- [定义]: 系统正常运行的时间比例。

$$\alpha = \frac{\text{mean-time-to-failure}}{\text{mean-time-to-failure} + \text{mean-time-to-repair}}$$

- 可靠性**——是指系统能够保持正常运行的能力。通常用MTTF来衡量。
- 构件技术:**
  - 安装冗余系统
  - 错误报告和处理
  - .....

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

## 功能性

- [定义]: 系统能够完成所期望的工作的能力。

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

Usability（易用性）

- 关注：对用户而言完成某个期望任务的容易程度和系统所提供的用户种类。
- 划分为：
  - 可学习性：使系统容易学习使用。
  - 有效地使用系统：系统如何提高用户的操作效率。
  - 将错误的影响降到最低：系统如何将用户所犯错误...
  - 使系统适应用户的需要：如何使用户的任务变得轻松
  - 提高自信和满意程度：如何使用户确信采取了正确的行动。
- 细节不是软件体系结构层次上的问题——错！
  - 例如：如果用户在工作中想取消某个操作，返回操作开始前系统所处的状态，就需要在体系结构中规划该能力。

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

直接归属系统的执行属性

- 可在运行时观察确定的系统质量属性

质量属性	体系结构策略
性能	系统内各组件的交互；功能划分以开发系统的并行性。
安全性	特殊组件，如安全内核或认证服务器。
可用性	采用冗余组件增加容错性； 控制组件的交互。
功能性	与其他质量属性的交互。
易用性	易用性是一个积极因素；获得正确的信息流； 效率与性能的关系

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

直接归属于系统的质量属性

- 不可通过观察系统运行确定的质量属性

质量属性	体系结构策略
可修改性	模块化，封装组件
可移植性	可移植性层面
可重用性	减少组件的耦合
可集成性	可兼容的交互机制；一致的组件接口；
可测试性	使用关系允许渐进开发 模块化，封装组件；使用关系允许渐进开发

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

Modifiability（可修改性）

- [定义]：能够进行快速修改并使修改代价尽可能低的能力。
- 3类
  - 局部变化
  - 非局部变化
  - 软件体系结构变化
- 变化的原因：
  - 功能的扩展或改变（可扩展性）
  - 删除不想要的功能
  - 适应新的操作环境（可移植性）
  - 结构的重新调整
- 有时也称为可维护性
  - 从软件体系结构的方面考虑，是一致的观点

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

可重用性（Reusability）

- [定义]：通常指要合理的设计系统，使系统的结构或其某些组件能够在未来的应用开发中重复使用。
- 可集成性
- 可修改性的一个特例

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

可集成性（Integrability）

- [定义]：独立开发的系统组件能够协同运行的能力。
- 取决于：
  - 组件的外部复杂性、他们的交互机制和协议及组件功能划分的清晰度。
  - 组件结构的定义是否完整、合理
  - 要集成的组件的功能及该功能与组件的新环境的关系。
- 互操作性是特殊的可集成性。
  - 可集成性表明了系统各组成部分互相协作的能力
  - 互操作性衡量的是一组部件与另一个系统协作的能力

北京航空航天大学软件工程研究所



软件体系结构

创建软件体系结构

### 可测试性 (Testability)

- [定义]: 指通过测试使软件表露出缺陷的容易程度。
  - 可观察性
  - 可控制性

北京航空航天大学软件工程研究所

软件体系结构

直接归属于系统的质量属性

软件体系结构

### 不可通过观察系统运行确定的质量属性

质量属性	体系结构策略
可修改性	模块化, 封装组件
可移植性	可移植性层面
可重用性	减少组件的耦合
可集成性	可兼容的交互机制; 一致的组件接口;
可测试性	使用关系允许渐进开发 模块化, 封装组件; 使用关系允许渐进开发

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

### 问题?

- 从设计师的角度看, 以上对系统质量属性的定义存在如下问题:
  - 不可操作。如称某个系统是可修改的是没有意义的。
  - 问题的讨论点属于那个质量属性不清楚。如系统故障是可用性问题, 还是安全性问题, 还是易用性问题?

北京航空航天大学软件工程研究所

软件体系结构

为了更好地描述质量属性

创建软件体系结构

### 质量属性场景

- 刺激源(Source of stimulus).
  - 某个生成该刺激的实体 (a human, a computer system, or any other actuator)
- 刺激(Stimulus )
  - 是当刺激到达系统时需要考虑的条件。
- 环境
  - 刺激在某些条件下发生。当刺激发生时, 系统可能处于过载, 或者正在运行等。
- 制品(Artifact )
  - 被刺激的制品。可以是整个系统, 也可以是系统的一部分。
- 响应
  - 指刺激到达后所采取的行动。
- 响应度量
  - 当相应发生时, 应能够以某种方式对其进行度量, 以对需求进行测试

北京航空航天大学软件工程研究所

软件体系结构

质量属性场景

创建软件体系结构

### 区分:

- 一般质量属性场景: 独立于系统, 很可能适合任何系统的场景
- 具体质量属性场景: 指适合正在考虑的某个特定系统的场景。

北京航空航天大学软件工程研究所

软件体系结构

质量属性场景

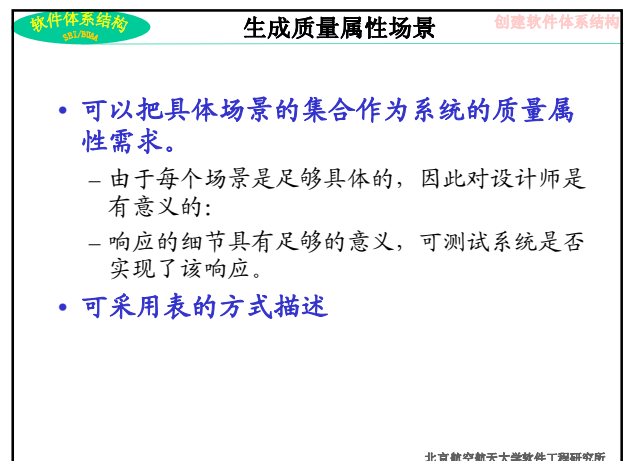
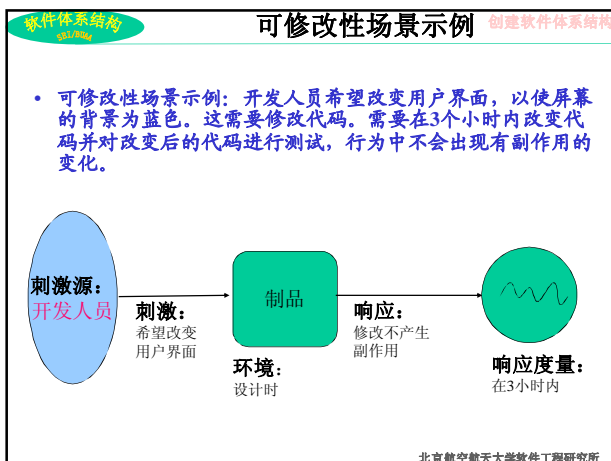
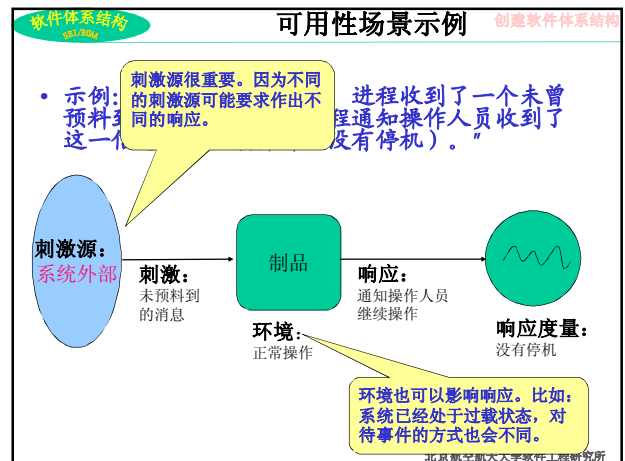
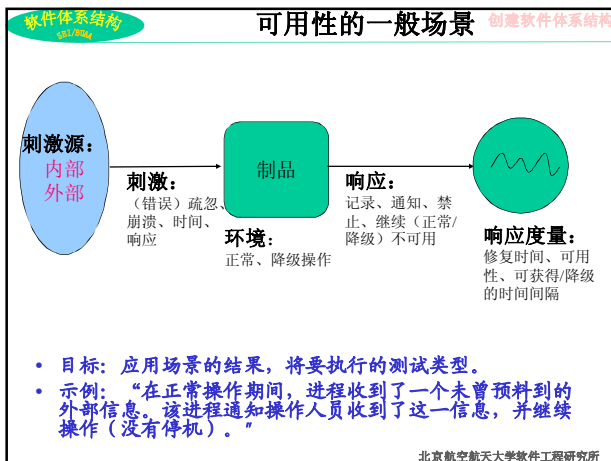
创建软件体系结构

### 质量属性场景的6个部分

```

graph LR
    A([刺激源]) -- 刺激 --> B[制品]
    B --- C[环境]
    B -- 响应 --> D(( ))
    D --- E[响应度量]
    
```

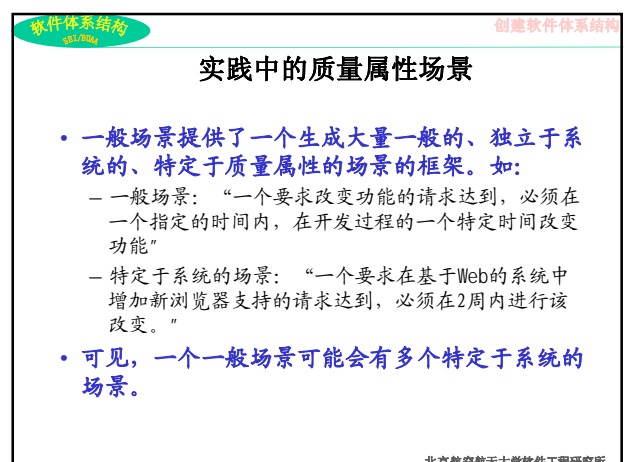
北京航空航天大学软件工程研究所



**软件体系结构** **例如: 可用性的一般场景生成** **体系结构**

场景	可能的值
源	系统内部、系统外部
刺激	错误: 疏忽、崩溃、时间、响应
制品	系统的处理器、通信通道、持久存储器、进程
环境	正常操作、降级模式
响应	系统应该检测事件, 并进行如下一个或多个活动 <ul style="list-style-type: none"> <li>将其记录下来</li> <li>通过适当的各方, 包括用户或其他系统</li> <li>根据已定义的规则禁止导致错误或事故的事件源</li> <li>在一段预先指定的时间间隔内不可用, 其中, 时间间隔取决于系统的关键程度</li> <li>继续在正常或降级模式下运行</li> </ul>
响应度量	系统必须可用的时间间隔 可用时间 系统可以在降级模式下运行的时间间隔 修复时间

北京航空航天大学软件工程研究所





软件体系结构

创建软件体系结构

- 下面讨论6个最常见的系统质量属性，
- 目的：
  - 确定质量属性描述所使用的概念；
  - 提供一个为该属性生成一般场景的方法

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

1、可用性（Availability）

- 可用性与系统故障及其相关后果有关。
  - 当系统不再提供其规范中所说明的服务时，就出现了系统故障。
  - 系统的用户（人或其它系统）可以观察到此类故障。

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

可用性

- 可用性所关注的方面：
  - 如何检测系统故障，
  - 系统故障发生的频度，
  - 出现故障时会发生什么情况，
  - 允许系统有多长时间非正常运行，
  - 什么时候可以安全地出现故障，
  - 如何防治故障的发生
  - 发生故障时要求进行哪些通知等
- 系统出现故障后，一个重要的概念
  - 修复该系统所需的时间：从出现故障到用户看不到故障的时间。

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

可用性

- 系统的可用性：系统正常运行的时间比例。
$$\alpha = \frac{\text{平均正常工作时间}}{\text{平均正常工作时间} + \text{平均修复时间}}$$

可得到结果，如99.9%的可用性，0.1%的故障率。
- 区分：故障和错误
- 故障和错误的区分为讨论自动修复策略提供了可能。
- 注意：预先约定的停机时间不是故障。

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

可用性的一般场景生成

场景	可能的值
源	系统内部、系统外部
刺激	错误：疏忽、崩溃、时间、响应
制品	系统的处理器、通信通道、持久存储器、进程
环境	正常操作、降级模式
响应	系统应该检测事件，并进行如下一个或多个活动 <ul style="list-style-type: none"> <li>将其记录下来</li> <li>通过适当的各方，包括用户或其他系统</li> <li>根据已定义的规则禁止导致错误或事故的事件源</li> <li>在一段预先指定的时间间隔内不可用，其中，时间间隔取决于系统的关键程度</li> <li>继续在正常或降级模式下运行</li> </ul>
响应度量	系统必须可用的时间间隔 可用时间 系统可以在降级模式下运行的时间间隔 修复时间

北京航空航天大学软件工程研究所

软件体系结构

创建软件体系结构

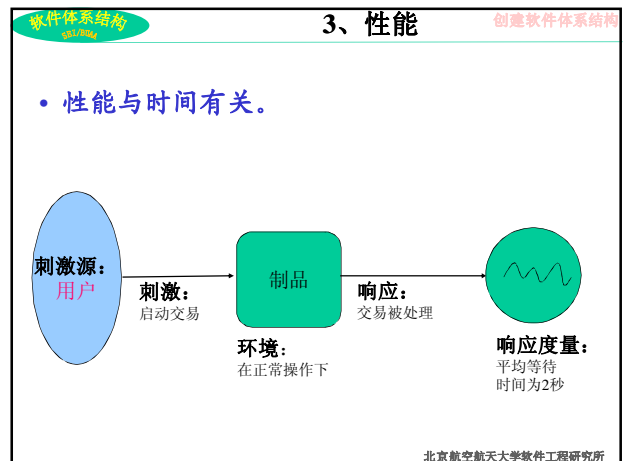
2、Modifiability（可修改性）

- 可修改性是有关变更的成本问题。
- 有两个关注点：
  - 可以修改什么（制品）？可以修改（增加、删除和修改）系统的任何方面，最常见的是：
    - 系统计算的功能
    - 系统存在的平台（硬件、操作系统和中间件等）
    - 系统运行的环境（系统必须与之交互操作的系统，系统用于与其它部分通信的协议等）
    - 系统所展示的质量属性（性能、可靠性，未来的可修改性等）
    - 系统容量（所支持的用户数量、同时发生的操作数量等）
  - 何时进行变更以及由谁进行变更（环境）？一旦指定了某个变更，就必须设计、实现、测试和部署新的实现。所有这些活动都需要时间和资金，这二者可以进行度量。

北京航空航天大学软件工程研究所

可修改性的一般场景生成	
<b>场景</b>	<b>可能的值</b>
<b>源</b>	最终用户、开发人员、系统管理员
<b>刺激</b>	希望增加/删除/修改/改变功能、质量数量、容器
<b>制品</b>	系统用户界面、平台、环境或与目标系统交互的系统
<b>环境</b>	在运行时、编译时、构建时、设计时
<b>响应</b>	查找体系结构中需要修改的位置，进行修改且不会影响其他功能，对所做的修改进行测试，部署所做的修改
<b>响应度量</b>	根据所影响的元素的数量度量的成本、努力、资金；该修改对其他功能或质量属性造成影响的程度

北京航空航天大学软件工程研究所



北京航空航天大学软件工程研究所

### 3、性能的一般场景生成

<b>场景</b>	<b>可能的值</b>
<b>源</b>	大量的独立源中的一个，可能来自系统内部
<b>刺激</b>	定期事件到达；随时事件到达；偶然事件到达
<b>制品</b>	系统
<b>环境</b>	正常模式；超载模式
<b>响应</b>	处理刺激；改变服务级别
<b>响应度量</b>	等待事件、期限、吞吐量、抖动、缺失率、数据丢失

北京航空航天大学软件工程研究所

### 4、安全性（Security）

- 安全性是衡量系统在向合法用户正常提供服务的情况下，阻止企图非授权使用和抗拒拒绝服务攻击的能力。
- 可以把安全性刻画为一个提供认可、机密性、完整性、保证、可用性和审核的系统。
  - 认可：交易（访问或修改数据或服务）不能被交易的任何一方拒绝的属性。如你通过Internet订购了一本书，就不能否认。
  - 机密性：未经授权不能访问数据或服务的属性。
  - 完整性：根据计划来提交数据或服务的属性。
  - 保证：交易的各方是所声称的人的属性。
  - 可用性：系统可用于合法使用的属性。
  - 审核：系统在其内部跟踪活动的属性。

北京航空航天大学软件工程研究所

### 安全性的一般场景生成

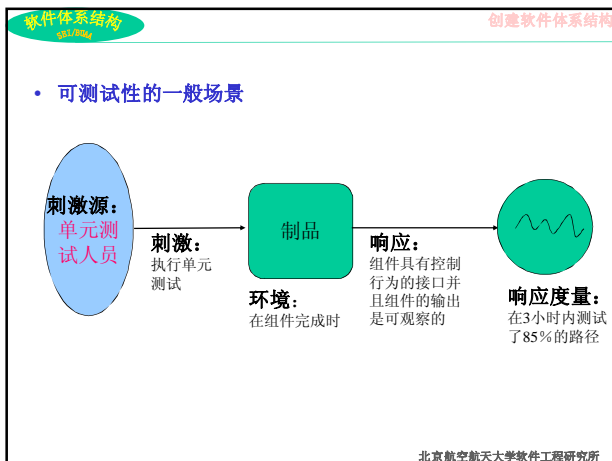
<b>场景</b>	<b>可能的值</b>
<b>源</b>	正确识别、非正确识别或身份未知的个人或系统，它来自内部/外部；经过了授权/未经授权。它访问了有限的资源/大量资源
<b>刺激</b>	试图显示数据、改变/删除数据、访问系统服务、降低系统服务的可用性
<b>制品</b>	系统服务、系统中的数据
<b>环境</b>	在线或离线、联网或断网、连接有防火墙或直接连接到了网络上
<b>响应</b>	对用户进行身份验证；隐藏用户的身份；阻止对数据和/或服务的访问；允许访问数据和/或服务；授予或收回对访问数据或服务的许可；根据身份记录访问/修改或试图访问/修改数据/服务；以一种不可读的格式存储数据；识别无法解释的对服务的高要求；通知用户或另外一个系统，并限制服务的可用性。
<b>响应度量</b>	用成功的概率表示、避免安全防范措施所需要的时间/努力/资源；检测到攻击的可能性；确定攻击或访问/修改数据和/或服务的个人的可能性；在拒绝服务攻击的情况下仍然可以获得的服务的百分比；恢复数据/服务；被破坏的数据/服务和/或被拒绝的合法访问的范围。

北京航空航天大学软件工程研究所

### 5、可测试性（Testability）

- 软件的可测试性：指通过测试揭示软件缺陷的容易程度。
  - 通常：测试占40%的开发成本
  - 假设：系统存在错误；
  - 能“控制”内部状态和输入，能“观察”输出

北京航空航天大学软件工程研究所



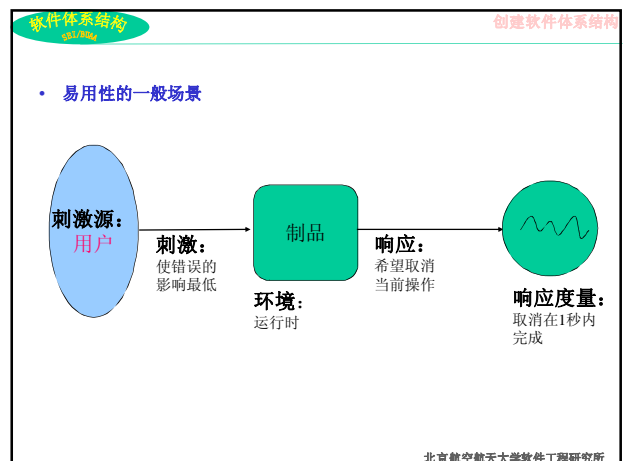
软件体系结构 创建软件体系结构

可测试性的一般场景生成

场景	可能的值
源	单元开发人员、增量集成人员
刺激	系统验证人员、客户验收测试人员、系统用户
制品	已完成的分析、体系结构、设计、类和子系统集成；所交付的系统设计、代码段、完整的应用
环境	设计时、开发时、编译时、部署时。
响应	提供对状态值的访问、提供所计算的值、准备测试环境
响应度量	已提供的可执行语句的百分比 如果存在缺陷出现故障的概率 执行测试的时间 测试中最长依赖链的长度 准备测试环境的时间

北京航空航天大学软件工程研究所

- 软件体系结构 创建软件体系结构
- 6、Usability（易用性）
- 关注：对用户而言完成某个期望任务的容易程度和系统所提供的用户种类。
  - 划分为：
    - 可学习性：使系统容易学习使用。
    - 有效地使用系统：系统如何提高用户的操作效率。
    - 将错误的影响降到最低：系统如何将用户所犯错误...
    - 使系统适应用户的需要：如何使用户的任务变得轻松
    - 提高自信和满意程度：如何使用户确信采取了正确的行动。
  - 细节不是软件体系结构层次上的问题——错！
    - 例如：如果用户在工作中想取消某个操作，返回操作开始前系统所处的状态，就需要在体系结构中规划该能力。
- 北京航空航天大学软件工程研究所



软件体系结构 创建软件体系结构

易用性的一般场景生成

场景	可能的值
源	最终用户
刺激	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
制品	系统
环境	在运行时或配置时
响应	<ul style="list-style-type: none"> <li>• 系统提供以下一个或多个响应来支持“学习系统特性”：帮助系统与环境联系紧密；界面为用户所熟悉；</li> <li>• 系统提供以下一个或多个响应来支持“有效使用系统”：数据和/或命令的聚合；已经输入的数据和/或命令的重用；支持在界面中的有效导航；具有一致操作的不同视图；全面搜索；多个同时进行的活动。</li> <li>• 系统提供以下一个或多个响应来“使错误的影响最低”：撤销；取消；从系统故障恢复；识别并纠正用户错误；检索忘记的密码；验证系统资源；</li> <li>• 系统提供以下一个或多个响应来“适配系统”：定制能力；国际化</li> </ul>
响应度量	任务时间、错误的数量、解决问题的数量、用户满意度、用户知识的获得、成功操作在总操作中所占的比例、北京航空航天大学软件工程研究所

北京航空航天大学软件工程研究所

软件体系结构 创建软件体系结构

使用一般场景进行沟通

- 一般场景的用途之一：和相关人员（涉众）进行沟通
- 质量属性刺激（有助于理解和发现问题）：

质量属性	刺激
可用性	不期望的时间、期望的事件不发生
可修改性	请求添加/删除/改变功能、平台、质量属性或容量
性能	周期性的、随即的或偶然的
安全性	试图显示数据、改变/删除数据、访问系统服务、降低系统服务的可用性
测试性	系统开发阶段的完成
易用性	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

北京航空航天大学软件工程研究所

## 软件体系结构和质量属性

我们将从以下三个方面讨论：

- 直接归属于系统的质量属性
  - 能通过观察系统运行确定的质量属性
  - 不能通过观察系统运行确定的质量属性
- 商业属性
- 与软件体系结构本身相关的质量属性

## 商业质量属性 (Business qualities)

- 也和系统质量属性一样，具有模糊性，也需要场景使其具体化、可测试。
- 商业质量属性
  - 开发成本或开发安排
    - 投放市场的时间：购买、重用
    - 成本和受益：不同的体系结构开发成本不同
    - 所希望的系统生命期的长短：质量属性需求不一样
  - 市场及营销策略
    - 目标市场：大众市场；专用系统——设计策略不一样。
    - 推出计划：如增量式发布，要求系统的扩展性好。
    - 与老系统的集成

## 软件体系结构的质量属性

- 软件体系结构的质量属性
  - 概念完整性
    - 在各个层次上统一系统设计的根本指导思想。
  - 正确性和完整性
    - 软件体系结构能够满足系统的各种需求及运行时的资源要求的必要条件。
  - 可构建性
    - 构建某个所希望的系统的难易程度。
    - 通常以成本和时间来衡量。

## 作业

- Trustworthy software Systems: A Discussion of Basic Concepts and Terminology
- 描述商业质量属性的场景，任选一个。
- 下次课前提交，课堂交流。