Chapter 1

1. architects职责：不用编码，权衡，指导

2. （ABC构架的商业周期）Software architecture is a result of technical, business, and social influences. Its existence in turn affects the technical, business, and social environments that subsequently influence future architectures. ABC是一个来自环境对架构以及反馈回环境的影响的循环。

3. 架构危险信号／现象（phenomena）：

（1）没有多个候选的设计方案

（2）构架依赖于某个版本的商业产品

（3）产生数据模块和使用数据的模块没分离

Chapter 2

1. 并发（concurrency）视图是属于组件－连接器（component－and－connecter）视图（课本p31）

Chapter 4

1. 可构建性（Buildability）属于构架的质量属性（课本p85）（Architecture Qualities）。

2.可构建性是什么：（课本P85）

可构建性是保证能够由指定的开发小组在规定的时间里及时开发系统，并允许在 开发过程中做某些更改的构架属性。最大程度上实现并发

（Buildability be completed by the available team in a timely manner and to be open to certain changes as development progresses）

3、商业质量属性：上市时间、成本和收益、系统生命期的长短、目标市场、推出计划、与老系统集成

与构架本身相关的质量属性：可构建性、概念完整性（conceptual integrity）

4. 老系统的集成（Integration with legacy systems）属于商业质量属性（Business Qualities）：必须分析现有系统与新系统集成对架构的要求。（课本P84）

5。质量属性场景（Quality Attribute Scenarios）是一种面向特定的质量属性的需求。（课本p66）

包括：刺激源（Source of stimulus），刺激（stimulus），环境（Environment ），制品（Artifact），响应（Response）和响应度量（Response measure）。

6。 涉众STAKEHOLDERS概念：对构建软件系统感兴趣的人。包括：客户，最终用户，开发人员，项目经理，维护人员以及那些对系统进行市场营销策划活动的人。（课本p6）不包括测试人员

Chapter 5

1. 心跳（heart beat）属于可用性战术（课本P89）

2. 可修改性战术（modifiability tactics）有几种方法：

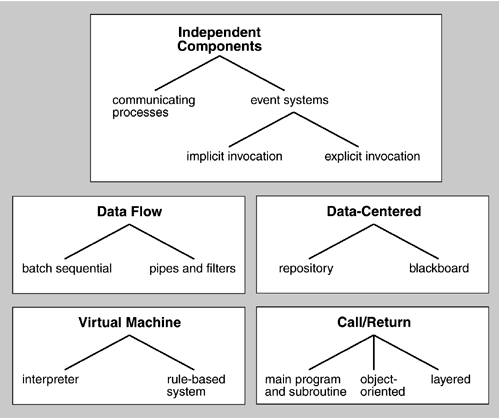
三种战术：局部化修改战术localize modifications、“防止连锁反应”战术prevent the ripple effect、推迟绑定时间（defer banding time）

1. 维持现有接口（Maintain existing interfaces）属于连锁反应／波纹效应（ripple effect）（p95）

添加接口，添加适配器，提供占位程序stub

5. 错误检测（fault detection）属于可用性战术（availability tactics）（p89）

6. 冗余（redundancy）属于错误恢复（fault recovery）（可用性）（p90）



编译器属于数据流构架模式

人工智能属于以数据为中心构架模式

Chapter 11

1. 数据延迟和交易吞吐量都是性能performance（p237）"data latency" and "transaction throughput."
2. risks（有风险决策）,nonrisks（无风险决策）, sensitivity points（敏感点）, and tradeoffs（权衡点）

## 1、互操作性10分（结合架构发展历史理解，看PPT）

关键词（**Interoperability）**

结合构架发展历史：

单系统、消息传递（面向对象体系构架）->中间件支撑->面向服务体系构架->模型驱动构架、面向语义服务体系架构；

构架之美提供更多的软件复用、降低耦合度（解耦）、提高互操作性，以解决质量问题、效率问题、互操作问题、业务灵活性问题。

互操作性——这条术语用来描述的是不同的程序借助于同一套交换格式来交换数据，读写相同文件格式以及采用相同协议的能力。

是指不同的计算机系统、网络、操作系统和应用程序一起工作并共享信息的能力。

* 访问互操作性（API）

（看上面两个图，理解一下访问互操作性咯，我大概的理解意思就是不同系统可以提供标准的API，用户可通过访问互操作性访问不同的系统，读写相同的文件格式，进行通讯和数据交换）

* 连接互操作性（SOAP）

（大概的理解是不同的客户端可利用不同标准的传输协议与服务器连接通讯就是数据交换，例如SOAP提供的就是连接互操作性）

* 语义互操作性（XML）

对于深入推进互操作性的任何工作而言，语义互操作性都不可或缺。如果一个系统能够进行通讯和交换数据，那么，它就具备语义协同工作能力。就数据的通讯而言，基本的要素包括规定的数据格式、通讯协议以及接口描述等等。一般而言，XML或SQL标准提供的就是语义互操作性。

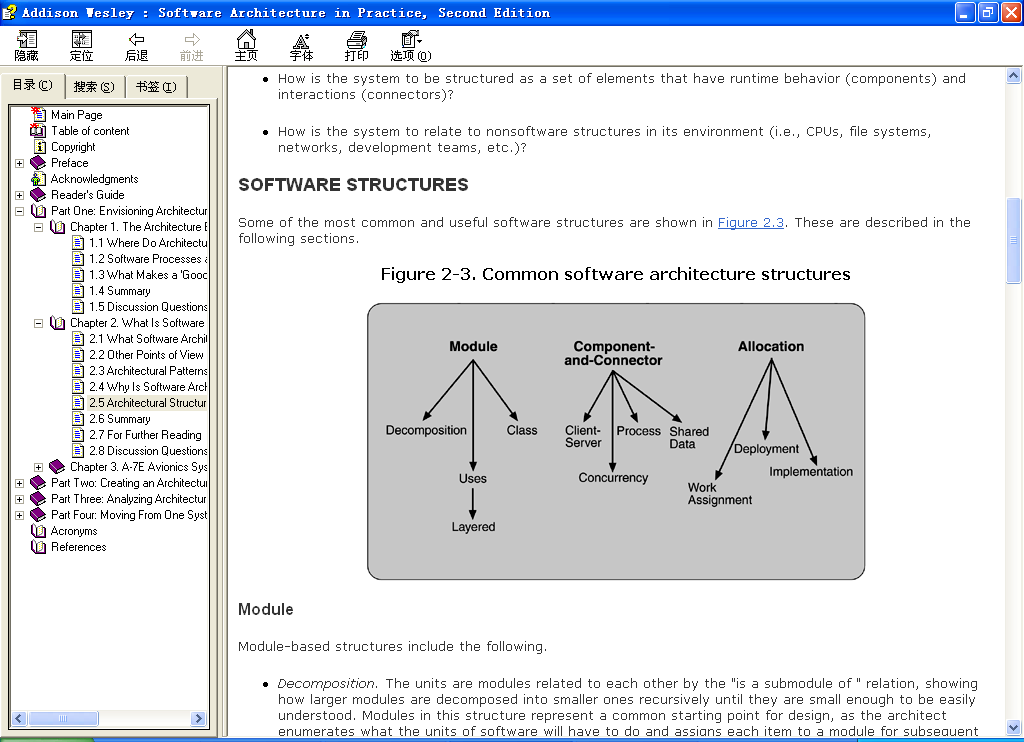
2、BATH是3个VIEW，（4+1View)

Bass 3 views:

模块：分解，使用，分层，类或泛化

组件连接器：进程或通信进程，并发，共享数据库或存储库，客户机-服务器。

分配：部署，实现，工作分配



4+1视图理论

**The logical view:** the classes, interfaces, and collaborations forming the vocabulary of the problem and its solution.

逻辑视图是一个模块视图

**The process view**: how control flows across the static classes at run time.

进程视图是一个组件连接器视图

**The implementation view:** the organization of the software modules in the development environment.

实现视图是一个分配视图

**The deployment view:** A mapping of elements onto processing and communication nodes.

部署视图是一个分配视图。

**Uses case view** cut across the views. They capture the system’s functional requirements for the system and help find conflicts.

3、分解和分层的区别：

分解：展示如何将较大的模块分解为较小的模块，能够容易理解。

分层：以一种特定的方式控制结构中的使用关系时，就出现了由层组成的系统，一个层是相关功能的一个集合。下一层使用上一层提供的服务。

4、评估阶段的步骤阶段是要背，简答题（P234~P242）

关键词（Steps of the evaluation phases）

评估阶段：

* First：ATAM方法的表述
* Second：商业动机的表述
* Third：构架的表述
* Fourth：对架构方法进行分类
* Fifth：生成质量属性效用树
* Sixth：分析构架方法（P238）
* Seventh：集体讨论并确定场景的优先级
* Eighth：分析构架方法
* Ninth：结构的表述

5、构建效用树的作用，拿来做什么。（P236）

关键词（Quality Attribute Utility Tree）

通过效用树（Utility Tree）的机制来对质量属性目标进行详细的清晰的阐述，其作用在于是质量属性需求具体化，从而迫使设计师和客户代表准确的定义出他们将要提供的相关质量属性。

“效用”是小效用树的根节点，“质量属性”构成该树的2级节点。性能、可修改性、安全性、易用性和可用性是效用树的子节点。

6、How to create a skeletal system?

* 首先实现处理架构组建的执行和交互的软件部分。这可能要求生成实时系统中的调度程序；实现规则引擎以控制在基于规则的系统中规则的激发；实现多进程系统中的进程同步机制或客户机。之后就拥有了一个运行系统。这为添加有用功能提供了基础。
* 现在可以选择把提供功能的那些元素添加到该系统。这种选择可能是根据以下因素做出的：首先处理问题最多的部分来降低风险；或现在开发人员的类型水平；或尽可能快地将有用的产品推向市场。
* 现在了提供下一个功能增量的元素后，就可以采用使用结构，以获知应该在系统中采用什么软件来支持该功能。
* 随着过程的继续，系统的增量越来越大，直到软件开发完毕。

7、Give an example for risk theme、sensitivity points

评估小组根据一些常见的基本问题或系统缺陷将风险分组为风险主题，从而增添了价值，例如，可以将文档编写不充分或者文档陈旧的若干个风险归为一个风险主题，即没有足够重视文档的编写。可以将关于系统不能再采用多种硬件是正常运行和/或软件故障归结为一个风险主题，即未提供备份能力或未提供高可用性。

敏感点：影响到质量属性

例如，同时数据库客户机的数量将影响数据库每秒钟能处理的交易的数量。因此，对于用每秒钟的交易数据量的响应来说，给服务器分配客户机就是一个敏感点。

1. 请根据所提供RFIDSale系统的架构模式和RFIDSale系统的需求规约简表RFIDSaleSRS，进行深入的分析，提出并明确系统的质量属性及其实现战术。
2. 在全面质量管理的思想指导下，根据系统的质量属性及其实现战术，提出系统的参考模型、参考的架构框架。
3. 在1,2的基础上，基于ADD方法，提出并详细描述目标系统RFIDSale的软件体系结构；
4. 基于ATAM方法，提出目标系统RFIDSale的软件体系结构的项目组内部评估结果。
5. 各位把六个质量属性和它们对应的战术都和这个系统的需求结合起来，扯上去，就可以得分。

例如:

安全性：PDA只能下单。需要保持数据的安全性，机密性。然后结合具体把安全性战术写上去。

例如：

互操作性：PDA如何和系统交互：PDA和系统交互是通过JMS的（涉及到SOA的知识），把互操作性扯上去。采用web service的方法

例如

性能：PC是通过Internet来和系统交互的。网络的带宽限制了性能。需要提高性能，这样可以使用.NET的缓冲池策略可以提高性能，（.net的缓冲池策略请参考南哥宝典442页上得PETshop）把想到的战术写上去。

1. 参考模型是MVC分层模型，是一个分布式-分层系统。PDA和服务器之间是采用了C/S的架构。PC和服务器是采用了B/S的架构。架构模式是C/S和B/S的混合架构模式。参考的架构框架采用了.net的MVC三层架构框架。
2. PDA和服务器之间是采用了C/S的架构。PC和服务器是采用了B/S的架构。结合ADD：先描述架构驱动因素，然后描述选择的架构。然后描述选择的架构，各位把上面说到的C/S和B/S的优缺点，写上去，并且说出C/S和B/S混合后的优点，.NET框架的优缺点。（C/S和B/S的优缺点请参考南哥宝典的140页之后描述的架构风格。），最后描述下分配的功能。
3. 第四题的话。在下实在不知道如何表达。各位看着办吧。