复习内容：

chapter 1--9, 11

软件体系结构（SA）概念及ABC【chapter1】

常见软件体系结构风格及其特点－数据流、调用/返回、层次、仓库/黑板

软件体系结构的多重视图、三重结构理论和架构描述（不包括ADL）

软件体系结构的质量属性（QA）及其战术（重点！）【chapter4&5】

QA：chapter4

可用性：故障（用户看的见的）VS错误（用户还未看见），修复时间（出现故障到用户看不见故障），可用性ɑ=平均正常工作时间/(平均正常工作时间+平均修复时间)

可修改性（一般场景）

性能（与时间有关：事件发生时，要耗费系统多长事件作出响应）到达模式（周期性、随机性、偶然）响应度量（等待时间、处理期限、系统吞吐量、响应抖动、缺失率or数据丢失）

安全性（向合法用户提供服务的同时，阻止非授权用户的使用的能力），攻击（试图突破防线），

可测试性（通过测试揭示软件缺陷的容易程度easy testing）、响应度量（测试中执行语句的百分比，如在3个小时内测试连85%的路径：、最长测试链的长度、发现额外的缺陷的可能性估计）

易用性（完成某个期望任务的程度和系统所提供的用户支持的种类）刺激（学习系统的特性、有效地使用系统、将错误的影响降到最低、使系统适应用户的需要、提高自信和满意度）

//other

商业质量属性：上市时间、成本和收益，所希望的系统生命期的长短，目标市场，推出计划，与老系统的集成

架构质量属性：概念完整性，正确性和完整性，可构建性

Chapter5：

战术：影响质量属性响应控制的设计决策。战术的集合：构架策略。

1可用性战术：（阻止错误发展成故障or把错误的影响限制在一定范围内）

1）错误检测（fault detection）： 命令/响应（ping/echo），心跳（heart beat （dead man timer）），异常（exceptions）

2）错误恢复（fault recovery）：

【准备恢复】：表决（voting），票死少数；主动冗余（Active Redundancy（hot restart）），分开多个路径放冗余，不会全挂；被动冗余（Passive Redundancy（暖启动/双冗余/三冗余）），确保备用状态是最新的，以便故障切换；备件（spare）要几分钟。

【修复系统】：shadow操作（shadow operation）恢复时模仿工作组件的行为；状态再同步（State Resynchronization）；检查点/回滚（Checkpoint /Rollback）

3）错误预防（fault Prevention）：从服务中删除（Removal from Service）；事务（Transaction），撤销整个绑定；进程监视器（Process monitor）为错误进程创建一个新的进程，初始化为某个状态。

2 可修改性战术

1. 局部化修改（Localize Modifications）：直接影响模块修改数量。维持语义的一致性（Maintain semantic coherence MSC）语义的一致性：模块中责任之间的关系，确保协同工作。“抽象通用服务”，如应用框架的使用和中间件的使用;预期期望的变更（Anticipate expected changes，AEC）使变更的影响降到最小，通常与语义一致性一起使用；泛化该模块（Generalize the module），模块越通用，越有可能通过调整语言而非修改模块来进行请求的变更；限制可能的选择（Limit possible options）。
2. 防止连锁反应（Prevent the ripple effect）：修改没有直接影响到的模块，因为依赖性。【8种类型的依赖性：语法（Syntax）、语义（Semantics）、顺序（Sequence）、A的一个接口的身份（Identity of an interface）、A的位置（运行时）（Location（runtime））、A提供的服务/数据的质量（Quality of service/data）、A的存在（Existence）、A的资源行为（Resource behavior）】信息隐藏（Hide Information）：隐藏信息将变更隔离在一个模块内；维持现有接口（Maintain existing interface）：创建能够屏蔽变化的抽象接口，该战术模式包括：【添加接口，原来接口不变；添加适配器，把A包装起来并提供原始A的签名；提供一个占位程序A】；限制通讯路径（Restrict communication paths）；仲裁者的使用（Use an intermediary）：管理依赖的活动：【仲裁者不能补偿语义变化，仲裁者是：数据（语法）、服务（语法）、A的接口身份、A的位置（runtime）、A的资源行为或由A控制的资源（runtime）、A的存在】。
3. 推迟绑定时间（Defer binding time）：运行时注册、配置文件、多态、组件更换、遵守已定义的协议。

3 性能战术（产生响应时间的两个基本因素：资源消耗和闭锁时间（资源争用、资源的可用性、对其他计算的依赖性））

1. 资源需求（Resource Demand）：（减少处理一个事件流所需要的资源：）提高计算效率；减少计算开销；（减少所处理的事件：）管理事件率；控制采样频率；（控制资源的使用：）限制执行时间；限制队列大小。
2. 资源管理（Resource Management）：引入并发；维持数据或计算的多个副本；增加可用资源。
3. 资源仲裁（Resource Arbitration）：先进/先出；固定优先级调用（语义重要性、时限时间单调、速率单调）；动态优先级调度（轮转、时限时间最早优先）；静态调度。

4 安全性战术（Security Tactics）

1. 抵抗攻击（Resisting Attacks）：对用户进行身份验证；对用户进行授权；维护数据的机密性；维护完整性；限制暴露的信息；限制访问。
2. 检测攻击（Detecting Attacks）：入侵检测。
3. 从攻击中恢复（Recovering from Attacks）：恢复：查看可用性；识别攻击者：审计追踪。

5 可测试性战术（Testability Tactics）

1. 输入/输出：记录/回放；将接口与实现分离；特化访问路线/接口。
2. 内部监控：内置监控器。

6 易用性战术：

1. 分离用户接口
2. 支持用户主动（取消，撤销，聚合）
3. 用户模型（用户模型；系统模型；任务模型）

软件体系结构设计的ADD方法（形成开发团队、创建骨架系统）

软件体系结构评审及其常用方法（ATAM）

案例分析（质量目标、如何达到、采用的架构方法）

考题类型

选择题（30%）

简答题（30%）

分析论述题（40%）