

Review Questions on Software Engineering

H3Art

他只给了1-11章的Q&A，后面的是我去找的，不一定用得上

- Review Questions on Software Engineering
 - Chapter 1: Introduction简介
 - Chapter 2: Software processes软件过程
 - Chapter 3: Agile Software Development敏捷软件开发
 - Chapter 4: Requirements Engineering需求工程
 - Chapter 5: System modeling系统建模
 - Chapter 6: Architectural design架构设计
 - Chapter 7: Design and implementation设计和实现
 - Chapter 8: Testing测试
 - Chapter 9: Evolution演进
 - Chapter 10: Sociotechnical systems社会技术系统
 - Chapter 11: Dependability and Security可靠性和安全性
 - Chapter 12: Dependability and security specification可靠性与安全规格
 - Chapter 16: Software reuse软件重用
 - Chapter 17: Component-based software engineering基于组件的软件工程
 - Chapter 22: Project Management项目管理
 - Chapter 23: Planning and Estimation计划与估计

Chapter 1: Introduction简介

What are the essential attributes of good software?

- Maintainability, dependability and security, efficiency and acceptability
- Run Effectively and Efficiently, Easy to be Extended, Easy to be Understood

好的软件应具备哪些基本属性？

- 可维护性、可靠性和安全性、效率和可接受性
- 有效和高效地运行，易于扩展，易于理解

What are the two fundamental types of software product?

- Generic products that are designed to meet the needs of many different customers
- Customised products designed to meet the specific needs of a single customer

有哪两种基本类型的软件产品？

- 通用产品：旨在满足许多不同客户的需求
- 定制产品：旨在满足单一客户的特定需求

What is software engineering?

- An engineering discipline concerned with all aspects of software production from specification to system maintenance

什么是软件工程？

- 一门工程学科，涵盖了软件产品从规格说明到系统维护的所有方面

What are the four fundamental activities in software processes?

- Software specification
- Software development
- Software validation
- Software evolution

软件过程中的四个基本活动是什么？

- 软件规格说明
- 软件开发
- 软件验证
- 软件演化

What is the distinction between computer science and software engineering?

- Computer science is concerned with theories and methods of computers and software systems; software engineering is concerned with the practice of software production

计算机科学与软件工程之间的区别是什么？

- 计算机科学涉及计算机和软件系统的理论和方法；软件工程涉及软件生产的实践过程

What are the 3 general issues that affect many different types of software?

- Heterogeneity. Software may have to execute on several different types of system
- Business and social change, which drives requirements for software change
- Security and trust - our software systems have to be secure against external and internal threats so that we can trust those systems

会影响许多不同类型软件的三个常见问题是什么？

- 异质性。软件可能需要在几种不同的系统上执行
- 商业和社会变化，推动对软件变更的需求
- 安全性和可靠性——我们的软件系统必须能够抵御内外部威胁，以便我们可以信任这些系统

List 5 different types of software application.

- Any 5 from stand-alone products
 - interactive transaction-based systems
 - embedded control systems
 - batch processing systems
 - entertainment systems
 - systems for modelling and simulation
 - data collection systems
 - systems of systems

列举5种不同类型的软件应用。

- 其中的任意五个皆可
 - 交互事务系统
 - 嵌入式控制系统
 - 批处理系统

- 娱乐系统
- 建模和仿真系统
- 数据采集系统
- 系统的系统

What software engineering fundamentals apply to all types of software systems?

- Systems should be developed using a managed and understood development process
- Dependability and performance are key system characteristics
- Understanding and managing the software specification and requirements are important
- Effective use should be made of available resources

哪些软件工程基本原则适用于所有类型的软件系统？

- 应使用可管理和已了解的开发过程开发系统
- 可靠性和性能是关键的系统特征
- 理解和管理软件规格说明和要求非常重要
- 应充分利用可用资源

What are three key characteristics of the engineering of web-based software engineering?

- Software reuse is the principal approach for constructing web-based systems
- Requirements for those systems cannot be completely specified in advance
- User interfaces are constrained by the capabilities of web browsers

网络软件工程的三个关键特征是什么？

- 软件重用是构建基于网络的系统的主要方法
- 对这些系统的要求不能完全事先规定
- 用户界面受网络浏览器的功能限制

What is a software engineering code of ethics?

- A set of principles that set out, in a general way, standards of expected behaviour for professional software engineers

什么是软件工程道德准则？

- 一组准则，以一般方式阐明了专业软件工程师的行为标准

Chapter 2: Software processes软件过程

What are the fundamental activities that are common to all software processes?

- Software specification
- Software design and implementation
- Software validation
- Software evolution

所有软件过程共同具有的基本活动是什么？

- 软件规格说明
- 软件设计和实现
- 软件验证
- 软件演化

List the 3 generic process models that are used in software engineering?

- The waterfall model
- Incremental development
- Reuse-oriented software engineering

列出在软件工程中使用的三种通用过程模型。

- 瀑布模型
- 增量开发
- 重用导向的软件工程

Why are iterations usually limited when the waterfall model is used?

- The waterfall model is a document-driven model with documents produced at the end of each phase. Because of the cost of producing and approving documents, iterations are costly and involve significant rework. Hence they are limited

当使用瀑布模型时，为什么迭代通常是有限制的？

- 瀑布模型是一个文档驱动模型，在每个阶段结束时生成文档。由于生成和批准文档的成本，迭代和成本高昂并且涉及大量返工。因此迭代通常是有限制的

What are the three benefits of incremental development, compared to the waterfall model?

- The cost of accommodating changes to customer requirements is reduced
- It is easier to get customer feedback on development work that has been done
- More rapid delivery and deployment of useful software to the customer is possible

与瀑布模型相比，增量开发的三个优点是什么？

- 减少适应客户需求变化的成本
- 更容易获取有关已完成开发工作的客户反馈
- 可以更快地向客户交付和部署有用的软件

What are the development stages in reuse-based development?

- Component analysis
- Requirements modification
- System design with reuse
- Development and integration

重用开发的开发阶段是什么？

- 组件分析
- 需求修改
- 具有重用功能的系统设计
- 开发和集成

What are the principal requirements engineering activities?

- Feasibility study
- Requirements elicitation and analysis
- Requirements specification
- Requirements validation

什么是需求工程的主要活动？

- 可行性研究
- 需求激发和分析
- 需求规格说明
- 需求验证

Why is it increasingly irrelevant to distinguish between software development and evolution?

- Few software systems are now completely new and a more realistic mode of software development is of an iterative process that lasts for the lifetime of the software

为什么越来越没有必要区分软件开发和演化？

- 很少有软件系统是全新的，更实际的软件开发模式是一个持续维护软件的迭代过程

What are the advantages of using incremental development and delivery?

- Early delivery of critical functionality to the customer
- Early increments serve as prototypes to explore requirements
- Lower risk of overall project failure
- More extensive testing of critical customer functionality

使用增量开发和交付的好处是什么？

- 早期将关键功能交付给客户
- 早期增量可用作探索需求的原型
- 整个项目失败的风险较低
- 可对重要客户功能进行更广泛的测试

What are the 4 sectors in each loop in Boehm's spiral model?

- Objective setting
- Risk assessment and reduction
- Development and validation
- Planning

Boehm的螺旋模型中每个循环中的4个部分是什么？

- 目标设定
- 风险评估和降低
- 开发和验证
- 规划

What are the six fundamental best practices in the RUP(Rational Unified Process)?

- Develop software iteratively
- Manage requirements
- Use component-based architectures
- Visually model software
- Verify software quality
- Control changes to software

RUP（同一软件开发过程）中的六个基本最佳实践是什么？

- 迭代开发软件
- 管理需求
- 使用基于组件的架构
- 可视化软件模型
- 验证软件质量
- 控制软件的变更

Chapter 3: Agile Software Development敏捷软件开发

What are the shared characteristics of different approaches to rapid software development?

- The processes of specification, design and implementation are inter-leaved
- The system is developed and delivered as a series of versions
- User interfaces are often developed using an interactive development system that supports rapid UI development

快速软件开发方法的不同方法共享哪些特征?

- 规范，设计和实现的过程交织在一起
- 系统作为一系列版本开发和交付
- 经常使用交互式开发系统开发用户界面，支持快速UI开发

For what types of system are agile approaches to development particularly likely to be successful?

- Small and medium-sized software product development.
- Custom software development in an organization where there is a clear commitment from customers to become involved in the development process

敏捷方法在哪些类型的系统中特别容易成功?

- 小型和中型软件产品开发。
- 在客户明确承诺参与开发流程的组织中进行定制软件开发

List the 5 principles of agile methods.

- Customer involvement
- Incremental delivery
- People not process
- Embrace change
- Maintain simplicity

列出敏捷方法的5个原则。

- 客户参与
- 增量交付
- 人而非流程
- 拥抱变化
- 保持简单

List 4 questions that should be asked when deciding whether or not to adopt an agile method of software development.

Any 4 from those below. Others are also possible(see Ch 3)

- Is an incremental delivery strategy realistic?

- What type of system is being developed?
- What is the expected system lifetime?
- How is the development team organized?
- Is the system subject to external regulation?
- How large is the system that is being developed?

决定是否采用敏捷软件开发方法时应提出哪四个问题？

从以下任选4个问题。也可能有其他问题（见第三章）

- 增量交付策略是否现实？
- 正在开发什么类型的系统？
- 预期系统寿命周期有多长？
- 开发团队是如何组织的？
- 系统是否受外部监管？
- 正在开发的系统有多大？

What are three important characteristics of extreme programming?

- Requirements expressed as scenarios
- Pair programming
- Test-first development

极限编程的三个重要特点是什么？

- 用场景表达需求
- 成对编程（两个程序员在同一台计算机上共同完成同一个需求的开发工作，其中一个程序员负责编写代码的任务，另一个程序员负责观察，即时检查代码是否符合要求、讨论问题并提出建议）
- 先写测试再开发

What is test-first development?

- When a system feature is identified, the tests of the code implementing that feature are written before the code. Tests are automated and all tests are run when a new increment is added to the system

什么是测试优先开发？

- 当确定一个系统功能时，在编写代码实现该功能之前，先编写实现该功能的代码的测试。测试是自动化的，添加系统更新时会执行所有的测试

What are the possible problems of test-first development?

- Programmers may take short-cuts when developing tests so that the system tests are incomplete
- Some tests can be difficult to write incrementally
- It is difficult to estimate the completeness of a test set

测试优先开发可能存在的问题是什么？

- 程序员可能会在编写测试时采取捷径，使得系统测试不完整
- 有些测试会难以逐步编写
- 很难估计测试集的完整性

Briefly describe the advantages of pair programming.

- It supports the idea of common ownership and responsibility for the code
- It serves as an informal code review process

- It helps support refactoring

简要描述成对编程的优点是什么？

- 它支持代码的共同所有权和责任的想法
- 其作为非正式的代码审查流程
- 有助于支持重构

What is a Scrum sprint?

- A short (3-4 weeks) planning unit in which work to be done is assessed, features are selected for development, the software is implemented and delivered to system stakeholders

什么是Scrum开发的冲刺？

- 以短期（3-4周）的计划单位，在其中评估要完成的工作，选择要开发的特性，实现软件并交付给系统利益相关者

What are the barriers to introducing agile methods into large companies?

- Project managers may be reluctant to accept the risks of a new approach
- The established quality procedures in large companies may be incompatible with the informal approach to documentation in agile methods
- The existing teams may not have the high level of skills to make use of agile methods
- There may be cultural resistance if there is a long history of plan-driven development in the company

在大型公司中引入敏捷方法的障碍是什么？

- 项目经理可能不愿意接受新方法的风险
- 大型公司现有的质量程序可能与敏捷方法中非正式的文档方法不兼容
- 现有团队可能没有足够高的技能来使用敏捷方法
- 如果公司长期实行计划驱动的开发，可能会存在文化上的阻力

Chapter 4: Requirements Engineering需求工程

What are user requirements and system requirements?

- User requirements are statements in a language that is understandable to a user of what services the system should provide and the constraints under which it operates
- System requirements are more detailed descriptions of the system services and constraint, written for developers of the system

用户需求和系统需求是什么？

- 用户需求是用用户理解的语言陈述系统应该提供什么服务及其操作约束
- 系统需求是为系统开发人员编写的更详细的系统服务及约束的描述

What is the distinction between functional and non-functional requirements?

- Functional requirements define what the system should do
- Non-functional requirements are not directly concerned with specific system functions but specify required system properties or place constraints on the system or its development process

功能需求和非功能需求有什么区别？

- 功能需求定义系统应该做什么

- 非功能需求不直接涉及特定系统功能，而是指定所需的系统属性或对系统或其开发过程施加约束

List 3 types of non-functional requirement.

- Product requirements, that specify or constrain the software's behaviour
- Organisational requirements, are general requirements derived from policies and procedures in the customer's organization
- External requirements, which cover all requirements derived from factors external to the system and its development process

列举三种非功能需求类型

- 产品需求，规定或限制软件行为
- 组织需求，是从客户组织的政策和程序中推导的通用需求
- 外部需求，涵盖系统及其开发过程外部因素衍生的所有需求

What is the software requirements document?

- The official document that defines the requirements that should be implemented by the system developers

软件需求文档是什么？

- 定义由系统开发人员应该实现的需求的官方文档

What is the distinction between the terms 'shall' and 'should' in a user requirements document, which is written in natural language?

- 'Shall' normally indicates a mandatory requirement
- 'Should' indicates a desirable but not essential requirement

自然语言用户需求文档中“将”和“应该”的区别是什么？

- “将”通常表示强制性要求
- “应该”表示理想的但不是必要的要求

What are the main advantages of using a standard format to specify requirements?

- All requirements have the same format so are easier to read
- The definition of form fields mean that writers are less likely to forget to include information
- Some automated processing is possible

使用标准格式规定需求的主要优点是什么？

- 所有需求具有相同的格式，因此更易于阅读
- 表单字段的定义意味着作者不太可能忘记包含信息
- 可以进行一些自动化处理

What are the principal stages of the requirements engineering process?

- Requirements elicitation and analysis
- Requirements specification
- Requirements validation

需求工程过程的主要阶段是什么？

- 需求获取和分析

- 需求规格
- 需求验证

Give 5 reasons why eliciting requirements is difficult

- Stakeholders don't know what they want
- Stakeholders use their own language that requirements engineers may not understand
- Stakeholder requirements may conflict
- Political factors may influence the system requirements
- The business environment may change during elicitation

列举5个引起需求获取困难的原因

- 利益相关者不知道他们想要什么
- 利益相关者使用自己的语言，需求工程师可能不理解
- 利益相关者的需求可能存在冲突
- 政治因素可能影响系统需求
- 需求获取期间业务环境可能变化

What should be included in a scenario?

- A description of what's expected when the scenario starts
- A description of the normal flow of events
- A description of what can go wrong and how to handle it
- Information about concurrent activities
- A description of the system state when the scenario finishes

场景应包含什么？

- 开始时场景预期的描述
- 正常事件流的描述
- 可能出现的问题及其处理方法的描述
- 有关并发活动的信息
- 场景完成时系统状态的描述

What is a use-case?

- A use-case identifies a typical interaction with a system and the actors(human or computer) involved in that interaction

什么是用例？

- 用例标识与系统交互的典型操作以及涉及该交互的参与者（人类或计算机）

What is ethnography and how is it used in requirements elicitation?

- Ethnography is an observational technique where an analyst spends a period of time observing work and noting how the participants carry out their tasks. It is particularly useful in identifying essential cooperation in work processes

什么是人类学，以及如何在需求获取过程中使用它？

- 人类学是一种观察技术，在该技术中，分析师会花一段时间观察工作并记录参与者如何执行其任务。它在识别工作流程中的必要合作关系方面特别有用

What checks should be applied during requirements validation?

- Validity checks
- Consistency checks
- Completeness checks
- Realism checks
- The verifiability of the requirements should be assessed

在需求验证过程中应用哪些检查？

- 验证检查
- 一致性检查
- 完整性检查
- 现实检查
- 应评估需求的可验证性

List three requirements validation techniques

- Requirements reviews
- Prototyping
- Test-case generation

列举三种需求验证技术

- 需求审阅
- 原型
- 测试用例生成

What is requirements management?

- The process of managing changes to requirements during requirements specification and after the system has gone into use

什么是需求管理？

- 在需求规格说明期间和系统投入使用后管理需求变更的过程

What are the stages in the requirements change management process?

- Problem analysis and change specification
- Change analysis and costing
- Change implementation

需求更改管理过程的阶段是什么？

- 问题分析和变更规格说明
- 变更分析和成本估算
- 变更实施

Chapter 5: System modeling系统建模

What perspectives may be used for system modelling?

- An external perspective
- An interaction perspective
- A behavioural perspective

- A structural perspective

哪些视角可用于系统建模？

- 外部视角
- 交互视角
- 行为视角
- 结构视角

What UML diagram types may be used to represent the essential features of a system?

- Activity diagrams
- Use case diagrams
- Sequence diagrams
- Class diagrams
- State diagrams

哪些UML图类型可用于表示系统的基本特征？

- 活动图
- 用例图
- 时序图
- 类图
- 状态图

What is described in a context model?

- The immediate external environment of the system defining the system's context and the dependencies that a system has on its environment. The context model shows what is outside of the system boundary

上下文模型中描述了什么？

- 系统的即时外部环境，定义系统的上下文以及系统对其环境的依赖性。上下文模型显示系统边界之外的内容

How are activity diagrams used in describing the context of use of a system?

- Activity diagrams may be used to describe the business processes in which the system is used and the other systems which are also used in these processes

活动图在描述系统使用背景时如何使用？

- 活动图可用于描述系统使用的业务流程以及在这些流程中也使用的其他系统

What are the principal components of a textual use-case description?

- The actors involved
- A description of the interactions
- The data that is exchanged
- The stimulus that triggers the use case
- The response of the system
- Comments and other information

文本用例描述的主要组成部分是什么？

- 涉及的参与者
- 交互的描述

- 交换的数据
- 触发用例的刺激
- 系统的响应
- 注释和其他信息

What is illustrated in a UML sequence diagram?

- A sequence of interactions between the actors in a system's environment and the objects in the system itself. The sequence of interactions describes the implementation of a system feature or function

UML时序图表示什么？

- 系统环境中的参与者与系统本身中的对象之间的一系列交互。交互时序描述系统特性或功能的实现

How is generalization(inheritance) used to simplify the models of a system with many similar objects?

- Assuming that the similar objects have attributes and methods in common, these common attributes and methods are associated with a 'super-class' which generalizes all of the objects sharing these attributes/methods. The specific object classes only declare the attributes/methods specific to that class and they inherit the general attributes/methods from the super-class

如何使用泛化（继承）来简化具有许多相似对象的系统的模型？

- 假设相似的对象具有相同的属性和方法，这些共同的属性和方法与“超类”相关联，该超类概括了共享这些属性/方法的所有对象。特定的对象类仅声明该类特定的属性/方法，并从超类继承通用的属性/方法

What is the basic assumption that underlies event-driven modelling?

- That the system can be represented as a model with a finite number of discrete states and external and internal events trigger a transition from one state to another.

事件驱动建模的基本假设是什么？

- 系统可以表示为具有有限数量的离散状态和外部和内部事件，这些外部和内部事件会触发从一个状态到另一个状态的转换

What are the claimed benefits of model-driven engineering?

- Engineers can work at a high level of abstraction without concern for implementation details
- Errors are reduced and the design and implementation process is speeded up
- By using powerful generation tools, implementations of the same system can be automatically generated for different platforms

模型驱动工程的声称的好处是什么？

- 工程师可以在高抽象度水平上工作，无需考虑实现细节
- 减少错误，加快设计和实现过程
- 通过使用强大的生成工具，可以自动为不同的平台生成相同系统的实现

What are the three types of abstract system model that are recommended by the MDA(Model Driven Architecture) method?

- A computation-independent model(CIM)
- A platform-independent model(PIM)
- One or more platform-specific models(PSMs)

什么是MDA（模型驱动架构）方法推荐的三种抽象系统模型类型？

- 计算无关模型(CIM)
- 平台无关模型(PIM)
- 一个或多个平台特定模型(PSMs)

Chapter 6: Architectural design架构设计

What are the advantage of explicitly designing and documenting a software architecture?

- It improves stakeholder communications
- It encourages a detailed analysis of the system
- It helps with large-scale reuse

明确设计和记录软件架构的优势是什么？

- 它改善了利益相关者沟通
- 它鼓励系统进行详细的分析
- 它有助于大规模重用

What are the two ways in which an architectural model of a system may be used?

- As a means of facilitating discussion about the most appropriate architecture for a system
- As a means of documenting the architecture of an existing or an intended system

系统的架构模型可能有哪两种用途？

- 作为促进关于系统最合适的架构进行讨论的手段
- 作为记录现有或预期系统架构的手段

List 4 fundamental questions that should be addressed in architectural design

- Is there a generic application architecture that can be used?
- How will the system be distributed?
- What architectural style or styles are appropriate?
- How should the system be structured?
- What control strategy should be used?

在架构设计中应解决的四个基本问题是什么？

- 是否存在可用的通用应用程序架构？
- 系统将如何分布？
- 哪些架构风格是合适的？
- 系统应该如何构建？
- 应该使用什么控制策略？

What are the fundamental architectural views proposed in Krutchen's 4+1 model?

- A logical view that shows the key abstractions of the system
- A process view that shows the interacting processes in the system
- A development view that shows how the system is decomposed for development
- A physical view that shows the distribution of software on the system hardware

Krutchen的4+1模型中提出的基本架构视图是什么？

- 逻辑视图，显示系统的关键抽象
- 流程视图，显示系统中相互作用的过程
- 开发视图，显示系统如何被分解为开发
- 物理视图，显示软件在系统硬件上的分布

What is an architectural pattern?

- A stylized abstract description of good practice in architectural design that has been tried and tested in different systems and environments. The pattern should include information on when it is and is not appropriate to use that architectural design

什么是架构模式？

- 对架构设计中良好实践的程式化抽象描述，已在不同系统和环境中进行了尝试和测试。该模式应包括有关何时适合或不适合使用该架构设计的信息

What is the fundamental characteristic of a repository architecture?

- All shared data is held in a central database that can be accessed by all sub-systems

仓库架构的基本特征是什么？

- 所有共享数据都保存在一个中央数据库中，所有子系统都可以访问

What is the most important advantage of a client-server architecture?

- This is a distributed architecture so that it is possible to provide services on different computers. It is easy to add a new server or upgrade existing servers without disrupting the system

客户端-服务器架构的最重要优势是什么？

- 这是一种分布式架构，因此可以在不同的计算机上提供服务。可以轻松添加新服务器或升级现有服务器而不会影响系统

Briefly describe pipe and filter architecture

- A system is decomposed into a set of functional transformations that consume inputs and produce outputs. Data flows from one function to another (the pipeline) and is transformed as it passes through the sequence

简要描述管道和过滤器架构

- 系统分解为一组功能变换，它们消耗输入并生成输出。数据从一个函数流到另一个函数（管道）并在通过序列时进行变换

What are transaction-processing applications?

- Database-centered applications that process user requests for information and update the information in the database. They are organized so that transactions cannot interfere with each other and the integrity of the database is maintained

什么是事务处理应用程序？

- 处理用户请求信息并更新数据库中的信息的以数据库为中心的应用程序。它们被组织起来，以便事务不会相互干扰并保持数据库的完整性

What are the principal functions of the 4 layers in a generic information system architecture?

- User interface
- User communications, authentication and authorization
- Information retrieval and modification
- Database and transaction management

通用信息系统架构的4层主要功能是什么？

- 用户界面
- 用户通信、身份验证和授权
- 信息检索和修改
- 数据库和事务管理

Chapter 7: Design and implementation设计和实现

What are the 5 key activities in an object-oriented design process?

- Understand and define the context and use of the system
- Design the system architecture
- Identify the principal objects in the system
- Develop design models
- Specify object interfaces

面向对象的设计过程中的五个关键活动是什么？

- 理解和定义系统的上下文和使用
- 设计系统架构
- 确定系统中的主要对象
- 开发设计模型
- 指定对象接口

What do you understand by the system context and interaction model?

- The system context is a static model of the other systems in the environment of the system being designed
- An interaction model is a dynamic model that describes how the system being designed interacts with its environment

系统上下文和交互模型是什么？

- 系统上下文是正在设计的系统环境中其他系统的静态模型
- 交互模型是一种动态模型，描述了正在设计的系统如何与其环境交互

Briefly describe 3 approaches that may be used to identify object classes?

- Grammatical analysis identifying nouns and verbs
- Identify tangible things in the application domain
- Use scenario-based analysis

简要描述可用于识别对象类的三种方法？

- 语法分析识别名词和动词
- 识别应用程序域中的有形事物
- 使用基于场景的分析

Why is it important to specify the interfaces of components that are being developed by a software engineering team?

- Interfaces have to be specified so that objects and sub-systems can be designed in parallel. Once an interface has been specified, the developers of other parts of the system may assume that the interface will be implemented

对于由软件工程团队开发的组件进行接口规格说明很重要，为什么？

- 必须指定接口，以便可以并行地设计对象和子系统。一旦指定了接口，系统的其他部分的开发人员可以假设该接口将被实现

What do Gamma et al. suggest are the four essential elements of a design pattern?

- A meaningful name
- A description of the problem and when the pattern can be applied
- A solution description, which shows the components in the solution and their relationships
- A statement of the consequences of applying the pattern

Gamm等人建议设计模式的四个基本元素是什么？

- 有意义的名称
- 问题描述以及何时可以应用该模式的描述
- 解决方案描述，显示解决方案中的组件及其关系
- 应用模式的后果声明

How do design patterns contribute to reuse?

- Patterns and pattern languages are ways to describe best practices, good designs and capture experience in a way that is possible for others to reuse

设计模式如何促进重用？

- 模式和模式语言是描述最佳实践、良好设计，以及以其他人可以重用的方式捕获经验的方法

What are the 4 levels at which software reuse is possible?

- The abstraction level where knowledge of successful abstractions is reused
- The object level where objects classes and methods from libraries are reused
- The component level where collections of objects are reused
- The system level where entire application systems are reused

软件重用有哪四个水平？

- 抽象层次，通过重用成功的抽象来重用
- 对象层次，通过库中的对象类和方法进行重用
- 组件层次，通过重用对象集合进行重用
- 系统层次，通过重用整个应用程序系统进行重用

What are the principal aims of software configuration management?

- To support system integration so that all developers can access the project code and documents in a controlled way, find out what components have been changed and compile and link components to create a system

软件配置管理的主要目标是什么？

- 支持系统集成，以便所有开发人员可以以受控的方式访问项目代码和文档，查找已更改的组件，并编译和链接组件以创建系统

What are essential tools in a software development platform?

- An integrated compiler and syntax-directed editing systems
- A language debugger
- Graphical editing tools for UML models
- Testing tools that can automatically run program tests
- Project support tools for code control

软件开发平台中的基本工具是什么？

- 集成编译器和语法指导的编辑系统
- 语言调试器
- 用于UML模型的图形编辑工具
- 可以自动运行程序测试的测试工具
- 用于代码控制的项目支持工具

Briefly describe the idea of open-source development.

- In an open-source development, the source code of a software system is made publicly available and volunteers participate in the further development of the system. Any contributor to an open source development may fix bugs and add new features to a system

简要描述开源开发的想法

- 在开源开发中，软件系统的源代码是公开可用的，志愿者参与到系统的进一步开发中。开源开发的任何贡献者都可以修复错误并为系统添加新功能

Chapter 8: Testing测试

What is the distinction between validation and verification?

- Verification: Are we building the product right?
- Validation: Are we building the right product?

验证和验证之间的区别是什么？

- 验证：我们是否正确地构建产品？（早期/过程验证）
- 验证/确认：我们是否在构建正确的产品？（末期/最终验证）

What are the advantages of inspections over testing?

- Inspections can discover many errors. In testing, one error may mask another
- Incomplete versions of a system can be inspected
- Inspections can consider broader quality attributes as well as program defects

检查比测试有哪些优点？

- 检查可以发现许多错误。在测试中，一个错误可能掩盖另一个错误
- 可以检查不完整的版本
- 检查可以考虑更广泛的质量属性以及程序缺陷

Briefly describe the three principal stages of testing for a commercial software system

- Development testing, where the system is tested to discover bugs and defects
- Release testing where the system is tested to check that it meets its requirements
- User testing where the system is tested in the user's environment

简要描述商业软件系统的三个主要测试阶段

- 开发测试，其中测试系统以发现错误和缺陷
- 发布测试，测试系统以检查是否符合要求
- 用户测试，其中系统在用户环境中进行测试

What tests should be included in object class testing?

- Tests for all operations in isolation
- Tests that set and access all object attributes
- Tests that force the object into all possible states

对象类测试中应包括哪些测试？

- 在隔离中测试所有操作
- 设置和访问所有对象属性的测试
- 强制对象进入所有可能的状态的测试

What guidelines does Whittaker suggest for defect testing?

- Chose inputs that force all error messages to be generated
- Design inputs that might cause buffers to overflow
- Repeat the same input numerous times
- Force invalid outputs to be generated
- Force computation results to be too large or too small

Whittaker为缺陷测试提出了哪些准则？

- 选择强制生成所有错误消息的输入
- 设计可能导致缓冲区溢出的输入
- 多次重复相同的输入
- 强制生成无效输出
- 强制计算结果过大或过小

What is an equivalence partition? Give an example.

- A class of inputs or outputs where it is reasonable to expect that the system will behave the same way for all members of the class. For example, all strings with less than 256 characters

什么是等价划分？可以举出一个例子吗？

- 一类输入或输出，可以合理地预期系统会对该类的所有成员以相同的方式进行处理。例如，所有字符长度小于256的字符串

What are the three important classes of interface errors?

- Interface misuse
- Interface misunderstanding
- Timing errors

接口错误的三个重要类别是什么？

- 接口误用
- 接口误解
- 定时错误

What should be the principal concerns of system testing?

- Testing the interactions between the components and objects that make up the system
- Testing reusable components and systems to check that they work as expected when integrated into the system

系统测试应该关注哪些主要方面？

- 测试组成系统的组件和对象之间的交互
- 测试可重用组件和系统以检查它们在整合到系统后的预期工作效果

Briefly summarize the test-driven development process

- Identify increment of functionality required
- Design tests for this functionality and implement as executable programs
- Run test along with other implemented tests. The test will fail
- Implement the functionality and re-run the test. Iterate until the test works
- Move on to implement the next chunk of functionality

简要概述测试驱动型开发过程

- 确定所需功能增量
- 为此功能设计测试，并将其实现为可执行程序
- 与其他实现的测试一起运行测试。测试失败了
- 实现功能并重新运行测试。反复到测试成功为止
- 接下来实现下一个功能块

What is scenario testing?

- Scenario testing is an approach to release testing where you write a story describing how a system may be used and design tests based on the sequence of events in the scenario

什么是场景测试？

- 场景测试是一种发布测试方法，您可以编写一个故事来描述系统的使用方式，并根据场景中的事件顺序设计测试

What is stress testing and why is it useful?

- Increasing the load on a system beyond its design limits to check how the system performs in this situation. This tests the failure behaviour of the system and may reveal defects that are only apparent when the system has to handle a heavy load

什么是压力测试，为什么有用？

- 将系统的负载提高到其设计限制之外，以检查系统在这种情况下的性能。这测试了系统的故障行为，并可能揭示仅在系统需要处理重负荷时才显现的缺陷

What are the three types of user testing?

- Alpha testing, where users work with the development team to test the software as it is being developed
- Beta testing where the software is released to selected users for testing before the formal system release
- Acceptance testing, where customers test a system to check that it is ready for deployment

用户测试有哪三种类型？

- Alpha测试，在此测试中，用户与开发团队一起测试正在开发的软件
- Beta测试，将软件发布给选定的用户进行测试，然后再进行正式系统发布

- 验收测试，客户测试系统以检查是否准备部署

Chapter 9: Evolution演进

Why is software evolution important?

- Organisations are completely dependent on their software systems and they are critical business assets. They must invest in evolution so that these systems remain useful and maintain their value

软件演进为什么很重要？

- 组织完全依赖于其软件系统，它们是关键业务资产。它们必须投资于演进，以使这些系统保持有用，并维持其价值

What are the stages in the system evolution process and what triggers that process?

The process is triggered by change requests. Process stages are:

- Impact analysis
- Release planning
- Change implementation
- System release

系统演进过程的阶段是什么，以及什么触发了该过程？

该过程由变更请求触发。过程阶段是：

- 影响分析
- 发布计划
- 变更实施
- 系统发布

Why might it sometimes be necessary to bypass the normal change management system and make urgent changes to a system?

- To repair a serious system fault
- To change the system to cope with unexpected changes in the system's operating environment
- To cope with unexpected business change

为什么有时需要绕过正常的变更管理系统并对系统进行紧急变更？

- 修复严重的系统故障
- 更改系统以应对系统运行环境的意外更改
- 应对意外的业务变化

What are Lehman's Laws and how were they derived?

- Lehman's 'laws' are a set of hypothesis which, it is claimed, set out invariants for system change. They were derived from studies of the growth and evolution of a number of large software systems

什么是Lehman定律，它们是如何推导出来的？

- Lehman的“定律”是一组假设，据称它们设置了用于系统变更的不变量。它们是根据对多个大型软件系统的增长和演变的研究推导出来的

What are the three different types of software maintenance and how is effort distributed across these maintenance types?

- Maintenance to repair software faults(17%)
- Maintenance to adapt the software to a different environment(18%)
- Maintenance to add to or modify the system's functionality(65%)

软件维护的三种不同类型是什么，这些维护类型的工作量分配如何？

- 修复软件故障维护（17%）
- 适应软件到不同环境维护（18%）
- 添加或修改系统功能的维护（65%）

What factors should be assessed to understand the relationship between a system and its environment?

- The number and complexity of system interfaces
- The number of inherently volatile system requirements
- The business processes in which the system is used

应该评估哪些因素以了解系统与其环境之间的关系？

- 系统接口数量和复杂性
- 内在不稳定的系统要求数量
- 系统使用的业务过程

What process metrics might be used to assess maintainability?

- Number of requests for corrective maintenance
- Average time required for impact analysis
- Average time taken to implement a change request
- Number of outstanding change requests

哪些过程指标可用于评估可维护性？

- 纠正性维护请求数
- 影响分析所需的平均时间
- 实现变更请求所需的平均时间
- 未完成变更请求的数量

What are the principal systems re-engineering activities?

- Source code translation
- Reverse engineering
- Program structure improvement
- Program modularisation
- Data re-engineering

主要的系统再造活动有哪些？

- 源代码翻译
- 反向工程
- 程序结构改进
- 程序模块化
- 数据再造

What are the strategic options for legacy system evolution?

- Scrap the system completely

- Leave the system unchanged and continue maintenance
- Re-engineer the system to improve maintainability
- Replace all or part of the system with a new system

传统系统演进的战略选项是什么？

- 完全废弃系统
- 保持系统不变并继续维护
- 对系统进行再造以提高可维护性
- 用新系统替换全部或部分系统

List four important factors used to assess applications for evolution.

Any four from:

- Understandability
- Documentation
- Data
- Performance
- Programming language
- Configuration management
- Test data
- Personnel skills

列出用于评估应用程序演进的四个重要因素。

选以下任意四项：

- 可理解性
- 文档
- 数据
- 性能
- 编程语言
- 配置管理
- 测试数据
- 人员技能

Chapter 10: Sociotechnical systems 社会技术系统

What is the difference between the business process layer and the organizational layer in the sociotechnical systems stack.

- The business process layer is concerned with the specific business processes that are used to support business functions
- The organizational layer is concerned with more general strategic issues such as business rules and compliance, organizational policies, etc

社会技术系统堆栈中的业务流程层和组织层之间有什么区别？

- 业务流程层涉及支持业务功能的具体业务流程
- 组织层则涉及更一般的战略问题，如业务规则和合规性、组织政策等

What is the difference between a technical and a sociotechnical system?

- Technical systems include hardware and software components but not procedures and processes. Sociotechnical systems are self-aware and include defined operational processes and procedures. People are an inherent part of sociotechnical systems

技术系统和社会技术系统之间有什么区别？

- 技术系统包括硬件和软件组件，但不包括程序和过程。社会技术系统具有自我意识并包含定义的操作过程和程序。人是社会技术系统不可分割的一部分

What are emergent properties?

- System properties that only become apparent when all of the system components have been integrated. In other words, properties that are characteristic of the system as a whole

什么是紧急性问题？

- 是指问题非常复杂，其相关实体数量众多，无法产生明确定义的问题规范。只有在开发解决方案时才能确定问题的真正本质

What are three influences on the reliability of a system?

- Hardware reliability
- Software reliability
- Operator reliability

影响系统可靠性的三个因素是什么？

- 硬件可靠性
- 软件可靠性
- 操作员可靠性

Why are sociotechnical systems non-deterministic?

- Partly because they include people whose behaviour may change from day to day and partly because changes to the hardware, software and data in these systems is so frequent that the consequences of these changes are impossible to ascertain

为什么社会技术系统是非确定性的？

- 部分是因为它们包括行为可能每天都在变化的人，部分是因为这些系统中的硬件、软件和数据变化如此频繁，以至于无法确定这些变化的后果

What is a wicked problem?

- A problem that is so complex with so many related entities that no definitive problem specification can be produced. The true nature of the problem only emerges when the solution is developed

什么是棘手的问题？

- 一个问题如此复杂，有如此多的相关实体，以至于无法产生明确的问题规范。问题的真正本质只有在制定解决方案时才会出现

What are the three principal stages of systems engineering?

- Procurement
- Development

- Operation

系统工程三个主要阶段是什么？

- 采购
- 开发
- 操作

What are the main drivers for system procurement decisions?

- The state of other organizational systems
- The need to comply with external regulations
- External competition
- Business re-organization
- Available budget

系统采购决策的主要驱动因素是什么？

- 其他组织系统的状态
- 需要遵守外部法规
- 外部竞争
- 业务重组
- 可用预算

Why are plan-driven(rather than agile) processes used in systems engineering?

- Because different parts of the system are developed by different groups at the same time and plans are needed to coordinate their activities

为什么在系统工程中使用计划驱动（而不是敏捷）流程？

- 因为系统的不同部分是由不同的小组同时开发的，需要计划来协调他们的活动

What are latent conditions and active failures?

- Latent conditions are vulnerabilities and weaknesses in a system that, at some stage, may contribute to system failure
- Active failures are some operational event or human error that triggers a sequence of events that could lead to system failure

什么是潜在条件和主动故障？

- 潜在条件是系统中的漏洞和弱点，在某个阶段可能会导致系统故障
- 主动故障是一些操作事件或人为错误，它会触发一系列可能导致系统故障的事件

Chapter 11: Dependability and Security可靠性和安全性

Give three reasons why a system's dependability is more important than its detailed functionality

- System failures affect a large number of people
- Users may reject systems that are unreliable, unsafe or insecure
- System failure costs may be very high
- Undependable systems may cause information loss

给出系统的可靠性比其详细功能更重要的三个原因

- 系统故障会影响大量的人
- 用户可能会拒绝不可靠、不安全或不安全的系统
- 系统故障成本可能非常高
- 不可靠的系统可能会导致信息丢失

What are the four principal dependability properties?

- Reliability
- Availability
- Safety(prevent damage to the outside)
- Security(Resist external attacks)

四个主要的可靠性属性是什么？

- 可靠性
- 可用性
- 安全性（防止对外部造成损害）
- 安全性（抵抗外部的攻击）

List two other system properties that are sometimes considered to be dependability properties

Any two from:

- Repairability
- Maintainability
- Survivability
- Error tolerance

列出另外两个有时被认为是可靠性属性的系统属性

任意两个来自：

- 可维修性
- 可维护性
- 生存能力
- 容错

Briefly define what availability means?

- Availability is the ability of a system to deliver services when requested
- The probability that a system will be up and running and able to deliver useful services to users at any given time

简要定义可用性的含义？

- 可用性是系统在请求时提供服务的能力
- 系统启动并运行并能够在任何给定时间向用户提供有用服务的可能性

Explain how a relatively unreliable system can provide a high level of availability

- Reliability is concerned with the correct delivery of system services
- Availability is concerned with the system's operational state. A system can be available but unreliable so long as system failures can be quickly detected and corrected before they affect the normal usage of the system

解释一个相对不可靠的系统如何提供高水平的可用性

- 可靠性与系统服务的正确交付有关
- 可用性与系统的运行状态有关。一个系统可以是可用但不可靠的，只要系统故障能够在它们影响系统的正常使用之前被快速检测和纠正

Explain the difference between a system fault and a system failure.

- A fault is an internal system condition that can lead to an erroneous system state
- A failure is an externally observed deviation from expected system behaviour

解释系统错误和系统故障之间的区别。

- 错误（fault）是可能导致错误系统状态的内部系统状况（一个系统组件或元素的状态发生了变化，这种变化可能会对系统的正常运行产生影响，但并不一定会导致系统的故障）
- 故障（failure）是从外部观察到的与预期系统行为的偏差（系统在执行某个功能时不能正确地完成，因此导致了系统故障）

What is the most important difference between the two classes of safety-critical system?

- In a primary safety-critical system, a failure can lead directly to an accident
- In a secondary safety critical system, a failure can lead to the introduction of faults into another system, whose failure can lead to an accident

这两类安全关键系统之间最重要的区别是什么？

- 在主要的安全关键系统中，故障会直接导致事故
- 在二级安全关键系统中，一个故障可能导致将故障引入另一个系统，而另一个系统的故障可能导致事故

What is the distinction between a hazard and an accident?

- A hazard is a condition of the system that has the potential to cause an accident
- An accident is an unplanned event or sequence of events that results in human death or injury or other damage to the system's environment

危险和事故有什么区别？

- 危险是系统的一种可能导致事故的情况
- 事故是导致人员死亡或受伤或对系统环境造成其他损害的计划外事件或事件序列

What are the three principal threats to the security of a system?

- Threats to the confidentiality of a system and its data
- Threats to the integrity of a system and its data
- Threats to the availability of a system and its data

系统安全的三大主要威胁是什么？

- 对系统及其数据机密性的威胁
- 对系统及其数据完整性的威胁
- 对系统及其数据可用性的威胁

What are the three controls may be put in place to enhance system security?

- Vulnerability avoidance
- Attack detection and neutralization
- Exposure limitation and recovery

可以实施哪些三个控制措施来增强系统安全性？

- 漏洞规避
- 攻击检测和中和
- 暴露限制和恢复

Chapter 12: Dependability and security specification 可靠性与安全规格

Describe the three types of dependability requirement?

- Functional requirements to define error checking and recovery facilities and protection against system failures
- Non-functional requirements defining the required reliability and availability of the system
- Excluding requirements that define states and conditions that must not arise

描述三种类型的可靠性要求？

- 定义错误检查和恢复设施以及防止系统故障的功能要求
- 非功能性需求定义了系统所需的可靠性和可用性
- 不包括定义不得出现的状态和条件的要求

What is risk-driven specification?

- Critical systems specification should be risk-driven
- This approach has been widely used in safety and security-critical systems
- The aim of the specification process should be to understand the risks (safety, security, etc.) faced by the system and to define requirements that reduce these risks

什么是风险驱动规范？

- 关键系统规范应该是风险驱动的
- 这种方法已广泛应用于安全和安保关键系统
- 规范过程的目的应该是了解系统面临的风险（安全、保障等）并定义降低这些风险的要求

What activities are part of the process of risk analysis?

- Risk identification
- Identify potential risks that may arise
- Risk analysis and classification
- Assess the seriousness of each risk
- Risk decomposition
- Decompose risks to discover their potential root causes
- Risk reduction assessment
- Define how each risk must be taken into eliminated or reduced when the system is designed

哪些活动是风险分析过程的一部分？

- 风险识别
- 识别可能出现的潜在风险
- 风险分析和分类
- 评估每个风险的严重性
- 风险分解
- 分解风险以发现其潜在的根本原因

- 风险降低评估
- 定义在设计系统时必须如何消除或减少每种风险

What are the three risk categories that are normally used in hazard assessment for safety-critical systems?

- Intolerable. Must never arise or result in an accident
- As low as reasonably practical(ALARP). Must minimise the possibility of risk given cost and schedule constraints
- Acceptable. The consequences of the risk are acceptable and no extra costs should be incurred to reduce hazard probability

安全关键系统的危害评估中通常使用的三个风险类别是什么？

- 无法忍受。绝不能发生或导致事故
- 尽可能低（ALARP）。考虑到成本和进度限制，必须最大限度地降低风险的可能性
- 可以接受。风险的后果是可以接受的，不应产生额外的成本来降低危险概率

What are the three possible strategies that can be used for risk reduction in safety-critical systems?

- Risk avoidance
- Risk detection and removal
- Damage limitation

可用于降低安全关键系统风险的三种可能策略是什么？

- 风险规避
- 风险检测和消除
- 损坏限制

Under what circumstances would you use Probability of Failure on Demand (POFOD) as a reliability metric?

- Appropriate for protection systems where services are demanded occasionally and where there are serious consequence if the service is not delivered
- Relevant for many safety-critical systems with exception management components

在什么情况下您会使用按需故障概率 (POFOD) 作为可靠性指标？

- 适用于偶尔需要服务且如果不提供服务会产生严重后果的保护系统
- 与许多具有异常管理组件的安全关键系统相关

List three types of functional reliability requirement

- Recovery requirements that are geared to help the system recover after a failure has occurred
- Redundancy requirements that specify redundant features of the system to be included
- Process requirements for reliability which specify the development process to be used may also be included

列出三种类型的功能可靠性要求

- 恢复要求旨在帮助系统在发生故障后恢复
- 冗余要求，指定要包含的系统的冗余功能
- 还可以包括指定要使用的开发过程的可靠性过程要求

List 4 types of security requirement?

- Safety problems are accidental - the software is not operating in a hostile environment. In security, you must assume that attackers have knowledge of system weaknesses

- When safety failures occur, you can look for the root cause or weakness that led to the failure. When failure results from a deliberate attack, the attacker may conceal the cause of the failure
- Shutting down a system can avoid a safety-related failure. Causing a shut down may be the aim of an attack
- Safety-related events are not generated from an intelligent adversary. An attacker can probe defenses over time to discover weaknesses

列出 4 种类型的安全要求？

- 安全问题是偶然的 - 软件没有在恶劣的环境中运行。在安全方面，您必须假设攻击者了解系统弱点
- 当发生安全故障时，您可以查找导致故障的根本原因或弱点。当故意攻击导致失败时，攻击者可能会隐瞒失败的原因
- 关闭系统可以避免与安全相关的故障。造成关闭可能是攻击的目的
- 安全相关事件不是由聪明的对手生成的。攻击者可以随着时间的推移探测防御以发现弱点

Why is assuring system security a more challenging problem than assuring system safety?

- Security requirements are more difficult to identify than safety requirements because a system attacker can use knowledge of system vulnerabilities to plan a system attack, and can learn about vulnerabilities from unsuccessful attacks

为什么确保系统安全比确保系统安全更具挑战性？

- 安全需求比安全需求更难识别，因为系统攻击者可以利用系统漏洞的知识来计划系统攻击，并且可以从不成功的攻击中了解漏洞

Suggest 4 reasons why formal specification is not widely used?

- Problem owners cannot understand a formal specification and so cannot assess if it is an accurate representation of their requirements
- It is easy to assess the costs of developing a formal specification but harder to assess the benefits. Managers may therefore be unwilling to invest in formal methods
- Software engineers are unfamiliar with this approach and are therefore reluctant to propose the use of FM
- Formal methods are still hard to scale up to large systems
- Formal specification is not really compatible with agile development methods

提出正式规范未广泛使用的 4 个原因？

- 问题所有者无法理解正式的规范，因此无法评估它是否准确地表达了他们的需求
- 评估开发正式规范的成本很容易，但评估收益却很难。因此，管理者可能不愿意投资于正式方法
- 软件工程师不熟悉这种方法，因此不愿意建议使用 FM
- 形式化方法仍然难以扩展到大型系统
- 形式化规范与敏捷开发方法并不真正兼容

Chapter 16: Software reuse软件重用

List the main benefits of software reuse?

- Increased dependability
- Reduced process risk
- Effective use of specialists
- Standards compliance
- Accelerated development

列出软件重用的主要好处？

- 增加可靠性
- 降低流程风险
- 有效利用专家
- 符合标准
- 加速开发

What are the main problems with software reuse?

- Increased maintenance costs
- Lack of tool support
- Not-invented-here syndrome
- Creating, maintaining, and using a component library
- Finding, understanding, and adapting reusable components

软件重用的主要问题是什么？

- 维护成本增加
- 缺乏工具支持
- 非此处发明综合症
- 创建、维护和使用组件库
- 查找、理解和调整可重用组件

What key factors should be considered when considering the most appropriate type of reuse?

- The development schedule for the software
- The expected software lifetime
- The background, skills and experience of the development team
- The criticality of the software and its non-functional requirements
- The application domain
- The execution platform for the software

在考虑最合适的再利用类型时应考虑哪些关键因素？

- 软件的开发进度
- 预期的软件生命周期
- 开发团队的背景、技能和经验
- 软件的关键性及其非功能性需求
- 应用领域
- 软件的执行平台

What are the three classes of application framework classes introduced in class?

- System infrastructure frameworks
- Middleware integration frameworks
- Enterprise application frameworks

课堂上介绍的应用框架类有哪三类？

- 系统基础架构
- 中间件集成框架
- 企业应用框架

What features are supported by most web application frameworks?

- Security
- Dynamic web pages
- Database support
- Session management
- User interaction

大多数 Web 应用程序框架都支持哪些功能？

- 安全
- 动态网页
- 数据库支持
- 会话管理
- 用户互动

List 4 types of specialization of software product lines?

- Platform specialization
- Environment specialization
- Functional specialization
- Process specialization

列出软件产品线的4种专业化类型？

- 平台专业化
- 环境专业化
- 功能专业化
- 流程专业化

What are the process steps involved in creating a new member of a software product line?

- Elicit stakeholder requirements
- Choose closest-fit family member
- Re-negotiate requirements
- Adapt existing system
- Deliver new family member

创建软件产品线新成员涉及哪些流程步骤？

- 引出利益相关者的要求
- 选择最合适的家庭成员
- 重新协商要求
- 调整现有系统
- 交付新的家庭成员

What are the main benefits of COTS reuse?

- Rapid deployment
- Easy to judge suitability
- Avoids development risks
- More focus on core activity
- Simplified updates

COTS 重用的主要好处是什么？

- 快速部署
- 容易判断是否适合
- 规避开发风险
- 更加关注核心活动
- 简化的更新

What is an ERP system?

- An Enterprise Resource Planning (ERP) system is a generic system that supports common business processes such as ordering and invoicing, manufacturing

什么是 ERP 系统？

- 企业资源规划 (ERP) 系统是一个通用系统，支持常见的业务流程，例如订购和开票、制造

What design choices have to be made when integrating COTS products to create a COTS-solution system?

- Which COTS products offer the most appropriate functionality?
- How will data be exchanged?
- What features of a product will actually be used?

集成 COTS 产品以创建 COTS 解决方案系统时必须做出哪些设计选择？

- 哪些 COTS 产品提供最合适的功能？
- 数据将如何交换？
- 产品的哪些功能将被实际使用？

Chapter 17: Component-based software engineering 基于组件的软件工程

What are the essential elements of component-based software engineering?

- Independent components
- Component standards
- Middleware
- A development process

基于组件的软件工程的基本要素是什么？

- 独立组件
- 组件标准
- 中间件
- 一个开发过程

What is Councill and Heineman's definition of a component?

- A software component is a software element that conforms to a component model and can be independently deployed and composed without modification according to a composition standard

Councill 和 Heineman 对组件的定义是什么？

- 软件组件是符合组件模型、按照组合标准无需修改即可独立部署和组合的软件元素

What are the two principal component interfaces?

- Provides Interface
- Requires Interface

两个主要组件接口是什么？

- 提供接口
- 需要接口

What is a component model?

- A component model is a definition of standards for component implementation, documentation and deployment

什么是组件模型？

- 组件模型是组件实现、文档和部署标准的定义

Give 6 examples of general services that might be implemented in component model middleware support

- Component Management
- Transaction management
- Resource management
- Interface definition
- Exception Management
- Component communications

给出 6 个可能在组件模型中间件支持中实现的通用服务示例

- 组件管理
- 交易管理
- 资源管理
- 接口定义
- 异常管理
- 组件通信

Give examples of changes you might make to a component to make it more reusable?

- Remove application specific methods
- Make names more general
- Make exception handling consistent for all methods
- Integrating required components to increase independence

举例说明您可能对组件进行的更改以使其更可重用？

- 删除应用程序特定的方法
- 让名字更通用
- 使所有方法的异常处理保持一致
- 集成所需组件以提高独立性

What are the stages of the CBSE process?

- Component acquisition
- Component management
- Component certification

CBSE 流程有哪些阶段？

- 组件获取
- 组件管理
- 组件认证

What is component composition? What are the three different types of composition?

- Component composition is the process of integrating components with each other, and with specially written 'glue code' to create a system or another component
- Sequential composition
- Hierarchical composition
- Additive composition

什么是成分组成？ 三种不同类型的组合是什么？

- 组件组合是将组件相互集成的过程，并使用专门编写的“粘合代码”来创建系统或另一个组件
- 顺序组合
- 层次组合
- 添加组合

Chapter 22: Project Management项目管理

What are important differences between software project management and other types of project management?

- The product(software) is intangible
- There are no standard software processes
- Large software projects are often one-off projects

软件项目管理和其他类型的项目管理之间有哪些重要区别？

- 产品（软件）是无形的
- 没有标准的软件流程
- 大型软件项目通常是一次性项目

List 4 fundamental project management activities

- Project planning
- Reporting, Risk management
- People management
- Proposal writing

列出 4 项基本项目管理活动

- 项目计划
- 报告、风险管理
- 人员管理
- 提案撰写

What are three related categories of risk?

- Project risks
- Product risks
- Business risks

三个相关的风险类别是什么？

- 项目风险
- 产品风险
- 商业风险

Suggest 4 risks that may threaten the success of a software project?

- staff turnover
- management change
- requirements change
- technology change
- hardware unavailability
- product competition
- specification delays
- size underestimate
- CASE tool under-performance

提出可能威胁软件项目成功的 4 种风险？

- 员工流动率
- 管理变革
- 需求变化
- 技术变革
- 硬件不可用
- 产品竞争
- 规格延迟
- 尺寸低估
- CASE 工具性能不佳

Give 2 examples of technology risks that may arise in a software project

- The system database cannot process as many transactions as expected
- Reused software components are defective

给出2个软件项目中可能出现的技术风险的例子

- 系统数据库无法处理预期数量的事务
- 重复使用的软件组件有缺陷

What is involved in risk monitoring?

- Regularly assessing the project risks to decide whether or not the risks are becoming more or less probable and whether the effects of the risks have changed

风险监控涉及哪些内容？

- 定期评估项目风险，以确定风险发生的可能性是否增大或减小以及风险的影响是否发生变化

What are the four critical factors in people management?

- Consistency
- Respect
- Inclusion

- Honesty

人员管理的四个关键因素是什么？

- 一致性
- 尊重
- 包容性
- 诚实

What are the different levels in the human needs hierarchy?

- Physiological needs
- Safety needs
- Social needs
- Esteem needs
- Self-realization needs

人类需求层次有哪些不同层次？

- 生理需要
- 安全需求
- 社会需求
- 尊重的需要
- 自我实现的需要

What factors might be considered when selecting people for a software development team?

- Domain experience
- Platform experience
- Programming language experience
- Problem solving ability
- Educational background
- Communication ability
- Adaptability
- Attitude
- Personality

为软件开发团队选择人员时可能会考虑哪些因素？

- 领域经验
- 平台体验
- 编程语言经验
- 解决问题的能力
- 教育背景
- 沟通能力
- 适应性
- 态度
- 性格

What are the key factors that influence the effectiveness of group communications?

- Group size
- Group structure

- Group composition
- The physical work environment
- The available communication channels

影响群组沟通有效性的关键因素有哪些？

- 团体规模
- 集团架构
- 团体构成
- 物理工作环境
- 可用的沟通渠道

Chapter 23: Planning and Estimation 计划与估计

What estimates have to be produced by software project managers?

- The effort required to complete each project activity
- The time required to complete each activity
- The total cost of each activity

软件项目经理必须做出哪些估算？

- 完成每项项目活动所需的努力
- 完成每项活动所需的时间
- 每项活动的总成本

What factors should be considered when deciding on a price for software?

- Market opportunity
- Cost estimate uncertainty
- Contractual terms
- Requirements volatility
- Financial health of the company developing the software

决定软件价格时应考虑哪些因素？

- 市场机会
- 成本估算的不确定性
- 合同条款
- 需求波动性
- 开发软件的公司的财务状况

What is included in a quality plan and a validation plan?

- Quality plan: The quality procedures and standards that should be used in a project
- Validation plan: The approach, resources and schedule used for system validation

质量计划和验证计划包含哪些内容？

- 质量计划：项目中应使用的质量程序和标准
- 验证计划：用于系统验证的方法、资源和时间表

What are the principal activities in the project scheduling process?

- Identify activities
- Identify activity dependencies
- Estimate resources for activities
- Allocate people to activities
- Create project charts

项目调度过程中的主要活动是什么？

- 确定活动
- 识别活动依赖性
- 估算活动资源
- 将人员分配给活动
- 创建项目图表

How are bar charts used in project scheduling?

- Bar-charts are a calendar based representation that show the duration of activities against time or show the times when individuals are assigned to work on particular activities against time. They show the expected start date, duration and end date of activities

条形图如何用于项目调度？

- 条形图是一种基于日历的表示形式，它显示活动的持续时间与时间的关系，或显示个人被分配从事特定活动的时间与时间的关系。它们显示活动的预计开始日期、持续时间和结束日期

What are the most important differences between agile planning and plan-based development?

- In plan-based development, a plan for the whole project is drawn up before the project starts and this plan is modified as more information becomes available during the project
- In agile planning, planning is iterative and only the next iteration of the software is planned, often in detail
- In plan-based development, the schedule is usually extended if problems occur; in agile planning, the system being developed is cut down so that the software can be released on time

敏捷规划和基于计划的开发之间最重要的区别是什么？

- 在基于计划的开发中，在项目开始之前制定整个项目的计划，并在项目期间随着更多信息的出现而修改该计划
- 在敏捷规划中，规划是迭代的，只有软件的下一次迭代才会被规划，而且通常是详细的
- 在基于计划的开发中，如果出现问题，工期通常会延长；在敏捷规划中，正在开发的系统被削减，以便软件能够按时发布

Briefly describe two types of cost estimation techniques?

- Experience-based techniques where the estimate is based on a manager's experience of past projects and the application domain
- Algorithmic cost modeling where a formulaic approach is used to estimate the development effort required, based on attributes of the software and the development team

简要描述两种类型的成本估算技术？

- 基于经验的技术，其中估算基于经理过去项目和应用领域的经验
- 算法成本建模，根据软件 and 开发团队的属性，使用公式化方法来估计所需的开发工作量

What is the generic formula used in algorithmic cost modeling to compute the effort required to develop some software?

- $\text{Effort} = A \times [\text{Size}^B] \times M$
- Size = size of program; B is a constant between 1 and 1.5 that reflects attributes of the product and M is a multiplier that takes project and process attributes into account

算法成本建模中用于计算开发某些软件所需工作量的通用公式是什么？

- 工作量 = $A \times [\text{Size}^B] \times M$
- Size=程序大小； B 是 1 到 1.5 之间的常数，反映产品的属性， M 是考虑项目和流程属性的乘数

What are the estimation models used in COCOMO II.?

- The application composition model
- The early design model
- The reuse model
- The post-architecture model

COCOMO II 中使用的估计模型是什么？

- 应用程序组合模型
- 早期设计模型
- 重用模型
- 后建筑模型

What are the four classes of attribute used to compute the multiplier M?

- Product attributes
- Computer(hardware) attributes
- Personnel attributes
- Project attributes

用于计算乘数 M 的四类属性是什么？

- 产品属性
- 计算机（硬件）属性
- 人员属性
- 项目属性