1、简单题

- **软件工程的定义**

  软件工程是一门研究用工程化方法构建和维护有效的、实用的和高质量的软件的学科。它涉及程序设计语言、数据库、软件开发工具、系统平台、标准、设计模式等方面。在 IEEE 中，软件工程是将系统化的、规范的、可度量的方法应用于软件的开发、运行和维护，即将工程化方法应用于软件。

  阅读经典名著“人月神话”等资料，解释 software crisis、COCOMO 模型。

  software crisis 即软件危机，是早期计算机科学的一个术语，是指在软件开发及维护的过程中所遇到的一系列严重问题，这些问题皆可能导致软件产品的寿命缩短、甚至夭折。软件开发是一项高难度、高风险的活动，由于它的高失败率，故有所谓“软件危机”之说。软件危机的本源是复杂、期望和改变。这个术语用来描述正急遽增加之电脑的力量带来的冲击和可能要处理的问题的复杂性。从本质上来说，它谈到了写出正确、可理解、可验证的计算机程序的困难。

  COCOMO，英文全称为 Constructive Cost Model，中文为结构性成本模型。它是由巴里 勃姆提出的一种软件成本估算方法。这种模型使用一种基本的回归分析公式，使用从项目历史和现状中的某些特征作为参数来进行计算。构造性成本模型由三个不断深入和详细的层次组成。第一层， “基本 COCOMO”，适用对软件开发进行快速、早期地对重要的方面进行粗略的成本估计，但因其缺少不同的项目属性（“成本驱动者”）的因素，所以准确性有一定的局限性。“中级 COCOMO”中考虑进了这些成本驱动者。“详细 COCOMO”加入了对不同软件开发阶段影响的考量。

- **软件生命周期。**

  软件生命周期（Software Development LifeCycle）是指软件的产生直到成熟的全部过程。其中软件生命周期模型是指人们为开发更好的软件而归纳总结的软件生命周期的典型实践参考。

- **按照 SWEBok 的 KA 划分，本课程关注哪些 KA 或 知识领域？**

  SWEBOK 的知识域包括软件需求（Software requirements）软件设计（Software design）软件建构（Software construction）软件测试（Software test）软件维护与更新（Software maintenance）软件构型管理（Software Configuration Management, SCM）软件工程管理（Software Engineering Management）软件开发过程（Software Development Process）软件工程工具与方法（Software Engineering Tools and methods）软件质量（Software Quality），

  本课程关注软件需求、软件设计、软件工程管理和软件开发过程

- **解释 CMMI 的五个级别。例如：Level 1 - Initial：无序，自发生产模式。**

Level 1——初始级 软件过程是无序的，有时甚至是混乱的，对过程几乎没有定义，成功取决于个人努力。管理是反应式的。

Level 2——已管理级 建立了基本的项目管理过程来跟踪费用、进度和功能特性。制定了必要的过程纪律，能重复早先类似应用项目取得的成功经验。

Level 3——已定义级 已将软件管理和工程两方面的过程文档化、标准化，并综合成该组织的标准软件过程。所有项目均使用经批准、剪裁的标准软件过程来开发和维护软件，软件产品的生产在整个软件过程是可见的。

Level 4——量化管理级 分析对软件过程和产品质量的详细度量数据，对软件过程和产品都有定量的理解与控制。管理有一个作出结论的客观依据，管理能够在定量的范围内预测性能。

Level 5——优化管理级 过程的量化反馈和先进的新思想、新技术促使过程持续不断改进。

- **用自己语言简述 SWEBok 或 CMMI （约 200 字）**

CMMI（Capability Maturity Model Integration）翻译称能力成熟模型集成，是一个过程改进方法，目的是帮助组织改进绩效，被用于引导横贯一个项目、一个部门或一个完整的组织的过程改进。在软件工程和组织发展中，CMMI 可以向组织提供用于有效的过程改进的基本元素，致力于三个区域：1.产品和服务开发(CMMI for Development) 2.服务创建、管理和交付(CMMI for Service) 3.产品和服务采购(CMMI for Acquisition)。CMMI 存在两种表现方式：continuous 和 staged，前者被设计为允许用户聚焦特定的、被认为对于企业眼下的商业目标而言非常重要的过程，后者同时提供了一个从 CMM 到 CMMI 的跃迁。总而言之，CMMI 就是用于帮助软件企业对软件工程过程进行管理和改进，增强开发与改进能力，从而能按时地、不超预算地开发出高质量的软件。

2、解释 PSP 各项指标及技能要求：

- **阅读《现代软件工程》的 PSP: Personal Software Process 章节。 http://www.cnblogs.com/xinz/archive/2011/11/27/2265425.html**

PSP 的特点：1、不局限于某一种软件技术 (如编程语言)， 而是着眼于软件开发的流程， 这样不同应用的工程师可以互相比较。 2、不依赖于考试，而主要靠工程师自己收集数据，然后统计提高。 3、在小型，初创的团队中， 高质量的项目需求很难找到，这意味着给程序员的输入质量不高，在这种情况下， 程序员的输出 (程序/软件)往往质量不高，然而这并不能全部由程序员负责。 4、PSP 依赖于数据。5、PSP 的目的是记录工程师如何实现需求的效率， 而不是记录顾客对产品的满意度。

- **按表格 PSP 2.1，了解一个软件工程师在接到一个任务之后要做什么，需要哪些技能，解释你打算如何统计每项数据？ （期末考核，每人按开发阶段提交这个表）**

PSP2.1	
Planning • Estimate Development • Analysis • Design Spec • Design Review • Coding Standard • Design • Coding • Code Review • Test Record Time Spent Test Report Size Measurement Postmortem Process Improvement Pla n	计划 • 估计这个任务需要多少时间 开发 • 分析需求 • 生成设计文档 • 设计复审 (和同事审核设计文档) • 代码规范 (为目前的开发制定合适的规范) • 具体设计 • 具体编码 • 代码复审 • 测试（包括自我测试，修改代码，提交修改） 记录时间花费 测试报告 计算工作量 事后总结 提出过程改进计划

对于每一项数据的统计，可以根据在每一项过程中所花费的时间数进行记录，以小时为单位制，记录在每一项进度的工作时长，其中要注意每一项进度的开始结束时间不相联，因此要分开记录。