

1. 软件危机的原因：软件自身规模的加大，复杂性提高，人们也尚未完全认识其本质和特点，缺乏有效的对于大型软件项目的管理方法，用户对软件需求的描述和软件开发人员对需求的理解存在差异，软件开发的技术人员和管理人员本身的能力问题。
2. 软件=程序+数据+文档，程序=算法+数据结构
3. 软件是计算机系统中与硬件相互依存的另一部分，它是包括程序，数据及其相关文档的完整集合
4. 质量的观点：超越的观点：质量是可以认识而不能定义的；用户的观点：质量是恰好达到目的；制造的观点：质量是与需求说明的一致；产品的观点：质量是与产品的内在特性相联系的；基于价值的观点：质量取决于顾客愿意支付的金额
5. 系统的方法：定义系统边界，确认活动和对象
6. 软件工程定义：工程化：1.“科学和数学的应用，通过这种应用将自然界中的物质属性和能源变得对人类有用”，工程化的基本概念是以可重复的一致的方式来解决问题的，软件工程是（1）将系统的，规范的，可度量的方法应用于软件的开发，运行和维护的过程。即工程化方法应用于软件。（2）在（1）中所述方法的研究
7. 软件工程与计算机科学的区别：计算机科学：研究计算机相关的理论，软件工程：把计算机技术看作工具，用以解决问题
8. 软件过程（生命周期）：是软件生命周期内为达到一定目标而必须实施的一系列相关过程的集合
9. 瀑布模型：软件开发过程与软件生命周期是一致的，相邻二阶段之间存在因果关系，需对阶段性产品进行评审。优点：软件生命周期模型，使软件开发过程可以在分析，设计，编码，测试和维护的框架下进行，开发过程具有系统性，可控性，克服了软件开发的随意性，缺点，项目开始阶段用户很难精确提出产品需求，修改需求十分普遍，项目开发晚期才能得到程序的运行版本，这时修改软件需求和开发中的错误代价很大，经常发生开发小组“堵塞状态”
10. V 模型：软件开发过程与测试的对应
11. 快速原型模型：采用基于构件的软件开发方法，尽量缩短软件开发周期，不宜采用过多的新技术，原型可抛弃也可不抛弃，优点：支持软件需求开发，产生正式文档，每个迭代循环都是线性过程。
12. RAD（快速应用开发）模型：侧重于短暂开发周期，缺点：需要足够多的人手，系统难以模块化，技术风险很高的情况下不宜采用该模型
13. 螺旋模型：线性模型+迭代.原型+系统化，螺旋模型适用于计算机软件整个生命周期，从螺旋中心开始启动，沿顺时针方向前进。优点：符合人们认识现实世界和软件开发的

客观规律，支持软件整个生命周期，保持瀑布模型的系统性，阶段性，开发者和用户共同参与软件开发。

14. 增量模型：增量：小而可用的软件，特点：在前面增量的基础上开发后面的增量，每个增量的开发可用瀑布或快速原型模型迭代的思路
15. 形式化方法：类似于瀑布模型的软件开发方法，开发过程是用形式化数学转换将系统描述转换成一个可执行程序。用严格的，数学的符号体系来规约，开发和验证基于计算机的系统，可以产生无缺陷软件的承诺，费时，昂贵，需要培训
16. 构建集成模型（专用过程模型）：融合了螺旋模型的特征，基于构件库和软件重用，
17. RUP:软件工程的过程，提供了在开发组织中分派任务和责任的纪律化方法。目标是在可预见的日程和预算前提下，确保满足最终用户需求的高质量产品（先启，精化，构建，产品化）
18. 项目的管理工作从项目分解开始
19. 项目管理：项目管理者在有限的资源约束下，运用系统的观点，方法和理论，对项目涉及的全部工作进行有效地管理
20. 体系结构的基线：
21. 项目分解的意义：项目组在项目实施期间要完成的工作或要开展的活动的一种层次性，树状的项目活动描述
22. WBS（任务分解结构图），用途：确定工作范围，配备人员，编制资源计划，监视进程，明确阶段里程碑，内容的验证
23. WBS 分解方法：基于可交付成果的划分，基于工作过程的划分
24. 三种常见的程序设计小组的组织形式：核心程序员制小组（高结构化），民主制小组（松散结构），层次式小组（高结构化和松散结构两者之间）
25. LOC：源代码的总行数 LOC 即为 NCLOC 和 CLOC 之和，利用 LOC 为指标进行估算
26. FP：功能点，根据事务信息处理程序的基本功能定义的，利用 FP 为指标进行估算。他的运动可以贯穿整个软件生命周期。是基于应用软件的外部，内部特性以及软件性能的一种间接的规模测量，功能点分析的一个主要的目标就是从用户的角度定义系统的能力
27. 功能点估算步骤：1.识别功能点的类型 2.识别待估算应用程序的边界和范围 3.计算数据类型功能点所提供的未调整的功能点数量 4.计算人机交互功能所提供的未调整的功能点数量 5.确定调整因子 6.计算调整后的功能点数量

28. 外部输入 (EI): 指一个基本处理, 它处理的是来自本应用边界之外的一组数据或者控制信息。外部输入的基本目的是为了维护一个内部逻辑文件 (ILF) 或者改变系统的行为
29. 外部输出 (EO): 指一个向应用边界之外发送数据或者控制信息的基本处理。外部输出的基本目的是为了向用户展示一组经过了除了提取之外的其他逻辑处理的数据或者控制信息
30. 外部查询 (EQ): 指一个向应用边界之外发送数据或者控制信息的基本操作。外部查询的基本目的是为了向用户展示提取的数据或者控制信息
31. ILF 内部逻辑文件: 指一组以用户角度识别的, 在应用程序边界内且被维护的逻辑相关数据或者控制信息, 主要目的是通过应用程序的一个或多个基本处理过程来维护数据
32. EIF 外部接口文件: 指一组在应用程序边界内被查询, 但它是在其他应用程序中被维护的, 以用户角度来识别的, 逻辑上相关的数据。主要目的是为边界内的应用程序提供一个或多个通过基础操作过程来引用的一组数据或信息
- 33.

FP计算

$$FP = CT * [0.65 + 0.01 * \sum Fi]$$

其中: CT为未调整功能点

Fi 是复杂性调节值

Fi 取值 0, 1, ..., 5

当 $Fi = 0$ 时, 表示 Fi 不起作用

$Fi = 5$ 时, 表示 Fi 作用最大

34. FP 优缺点: 优点: 与程序设计语言无关, 软件项目开发初期就能基本上确定系统的输入, 输出等参数, 功能点度量能用于软件项目的开发初期, 缺点: 涉及到的主观因素比较多, 信息领域中的某些数据有时不易采集, FP 的值没有直观的物理意义
35. 成本估算: 专家判定, 算法模型, 机器学习
36. COCOMO 模型: 构造性成本模型, 按详细程度分为: 基本模型, 中间模型和详细模型, 将软件项目类型划分成三类: 组织型, 嵌入型, 半独立型
37. 评估模型: MMRE (相对误差平均值), Pred: 估计在实际值 x%范围内的项目的百分比

38. 可靠性估算：错误植入法，分别测试法，软件平均故障间隔时间估算
39. 甘特图：也称时间表，用来建立项目进度表，在甘特图中，每项任务的完成以必须交付的文档和通过评审为标准，因此在甘特图中，文档编制与评审是软件开发进度的里程碑
40. 进度编制的常见技术：关键路径法(CPM),包括正推法和逆推法,计划评审技术(PERT),图形评审技术(GERT),CPM是根据指定的网络顺序逻辑关系和单一的历时估算,计算每一个活动的单一的,确定的最早和最迟开始和完成日期;PERT中,近似用三时估计法估算出三个时间值,即最短,最长和最可能持续时间,再加权平均算出一个期望值作为工作的持续时间
41. 风险：在给定情况下和特定时间内,那些可能发生的结果与预期结果之间的差异,差异越大,风险越大
42. 风险管理就是识别评估风险,建立,选择,管理和解决风险的可选方案和组织方法,包括风险标识,风险预测,风险评估和风险管理 with 监控四个活动
43. 软件质量评价特性：正确性,可靠性,易用性,效率,可维护性,可移植性
44. 软件质量控制：采用技术手段保证软件质量,组织技术评审,加强软件测试,推行软件工程标准,对软件的修改,变更进行严格控制,对软件质量进行度量
45. 软件质量保证：管理层的眼睛
46. QC：品质控制,产品的质量检验,发现质量问题后的分析,改善和不合格品控制相关人员的总称,QA：品质保证,通过建立和维持质量管理体系来确保产品质量没有问题
47. 项目进度计划步骤：进度编制,资源调整,成本预算,计划优化调整,形成基线计划
48. 软件配置管理的目标：标识变更,控制变更,确保变更,报告变更
49. 敏捷联盟宣言联盟：个体和交互胜过过程和工具,可以工作的软件胜过面面俱到的文档,客户合作胜过合同谈判,响应变化胜过遵循计划
50. 常见的敏捷过程有：SCRUM,Crystal,ADP,XP,其中XP(极限编程)是最著名的一个,由一系列简单却相互依赖的实践组成,这些实践结合在一起形成了一个敏捷开发过程。
51. 极限编程实践：完整团队,计划游戏(把一次迭代发布的内容称为planning game,在迭代之处确认阶段称为Iteration planning game,确认可发布内容范围称为“release planning game”),客户测试,简单设计,结对编程,测试驱使开发,改进设计,持续集成,集体代码所有权,系统隐喻,可持续的速度,编码标准

52. 计划游戏：初始探索，发布计划，迭代计划，任务计划，迭代，结论

53. 测试驱动开发：测试驱动的开发方法，验收测试，结论