

软件工程第二章

1. 软件生命周期

- a. 定义：软件产品或软件系统从设计、投入使用到被淘汰的全过程
- b. 一个典型的软件开发周期：用户需求 -> 需求说明书 -> 设计说明书 -> 源程序 -> 运行结果 -> 用户

2. 软件过程：问题定义 -> 技术开发 -> 方案集成 -> 目标现状

- a. 定义：软件工程定义了软件生产的一系列活动，这些活动贯穿于软件开发的整个过程
- b. 什么是软件过程？沟通、计划、建模、构造、部署
- c. 三个流派：CMM-能力成熟度模型、ISO 9000质量标准体系、ISO/IEC 15504(SPICE)-信息软件过程评估
- d. CMM-能力成熟度模型：
 - i. 定义：软件能力成熟度模型是一种对软件组织在定义、实施、度量、控制和改善其软件过程的实践中 各个发展阶段的描述形成的标准
 - ii. 等级 除虫定量优
 - 初始级
 - 可重复级
 - 已定义级
 - 量化管理级
 - 优化级

3. 软件过程模型

- a. 定义：软件开发全部过程、活动和任务的结构框架。它能直观的表达软件开发的全过程，明确规定要完成的主要活动、任务和开发策略。软件过程模型也常称为软件开发模型、软件生存周期模型、软件工程范型。
- b. 常用过程模型：瀑布模型、增量过程模型（增量模型、RAD模型）、演化过程模型（原型模型、螺旋模型）、喷泉模型、其他过程模型。

4. 瀑布模型：

- a. 定义：规定了各项软件工程活动，以及它们自上而下，相互衔接的固定次序，如瀑布流水，逐级下落；是一种使用广泛的、以文档为驱动力的模型；也称经典的生命周期模型。
- b. 瀑布模型的特点
 - 阶段间具有顺序性和依赖性。
 - 推迟实现的观点。
 - 每个阶段必须完成规定的文档，每个阶段结束前完成文档审查，及早改正错误。
- c. 瀑布模型的优点
 - 它提供了一个模板，使得分析、设计、编码、测试与维护工作可以在该模板的指导下有序展开，避免了软件开发、维护过程中的随意状态。
 - 对于需求确定、变量相对更少的项目，线性顺序模型仍然是一种可以考虑采取的过程模型。采用这种模型，曾成功进行过许多大型软件工程的开发
- d. 瀑布模型的缺点：线性过程太过理想化
 - 各个阶段的划分完全固定，阶段之间产生大量的文档，极大地增加了工作量。
 - 由于开发模型是线性的，用户只有等到整个过程的末期才能见到开发成果，从而增加了开发的风险。
 - 早期的错误可能等到开发后期的测试阶段才能发现，进而带来严重的后果。

e. 适用场景：瀑布模型适用于系统需求明确、技术成熟、工程管理较严格的场合。

5. 增量过程模型

a. 定义

- 增量过程模型是一种非整体开发的模型，是一种进化式的开发过程。
- 它允许从部分需求定义出发，先建立一个不完整的系统，通过测试运行这个系统得到经验和反馈，进一步使系统扩充和完善。如此反复进行，直至软件人员和用户对设计的软件系统满意为止。
- 包括增量模型和RAD

b. 增量模型

- i. 定义：结合了原型模型的基本要素和迭代的特征，采用了基于时间和线性序列，每个确定线性序列都会输出该软件的一个“增量”。
- ii. 特点
 - 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的模型；迭代的思路。
- iii. 优点
 - 增量包概念的引入，以及它不需要提供完整的需求。只要有一个增量包出现，开发就可以进行。
 - 在项目的初始阶段不需要投入太多的人力资源。
 - 增量可以有效地管理技术风险。
- iv. 缺点：每个增量必须提供一些系统功能，这使得开发者很难根据客户需求给出大小适合的增量。
- v. 适用场景：适用于软件开发中需求可能变化，具有较大风险，希望可以更早进入市场的软件工程。

c. 快速应用开发模型（RAD）

i. 定义

- 是一个增量过程模型，强调短暂的开发周期。
- RAD模型是瀑布模型的“高速变体”，通过基于组件的构建方法实现快速开发。如果需求以及项目范围得到明确界定，RAD能使开发团队在很短的时间内建立一个“全功能系统”。

ii. 缺点：

- 对大型项目而言，RAD需要足够的人力资源。
- 开发者和客户都要实现承诺，否则将导致失败。
- 并非所有系统都适合（不能合理模块化的系统、高性能需求并且要调整构件接口的、技术风险很高的系统均不适合）

iii. 适用场景：同增量模型

6. 演化模型：原型模型和螺旋模型

a. 特点：首要实现软件最核心的、最重要的功能

b. 原型模型

- i. 适用场景：客户定义一个总体目标集，但他们并不清楚系统具体的输入输出；或者开发者不确定算法的效率、软件与操作系统是否兼容以及客户与计算机交互的方法。
- ii. 缺点
 - 设计者在质量和原型间有所折衷
 - 客户意识不到一些质量问题
- iii. 优点
 - 克服了瀑布模型的缺点、减少了由于软件需求不明确带来的开发风险，出现需求变更，能迅速做出调整并没有太大的困难
 - 开发工具先进，开发效率高，成本低

c. 螺旋模型

i. 特点

- 与RAD模型相似，该模型结合了瀑布模型和原型模型的特点
- 螺旋模型强调风险管理，因此该模型适用于大型系统的开发

ii. 活动

螺旋模型沿着螺线旋转，在笛卡尔坐标的四个象限上分别表达了四个方面的活动：

- 制定计划。确定软件目标，选定实施方案，弄清项目开发的限制条件。
- 风险分析。分析所选方案，考虑如何识别和消除风险。
- 实施工程。实施软件开发。
- 客户评估。评价开发方法，提出修正建议。

iii. 优点

- 支持用户需求的动态变化
- 原型可看作形式的、可执行的需求规格说明，易于为用户和开发人员共同理解，还可以作为继续开发的基础，并为用户参与所有关键决策提供了方便
- 螺旋模型特别强调原型的可扩充和可修改性，原型的进化贯穿整个软件生存周期，这将有助于目标软件的适应能力
- 螺旋模型为项目管理人员及时调整管理决策提供了方便，进而降低开发风险。

iv. 缺点

- 如果每次迭代的效率不高，致使迭代次数过多，将会增加成本并推迟提交时间
- 使用该模型需要有相当丰富的风险评估经验和专门知识，要求开发队伍水平较高

v. 适应场合

- 支持需求不明确，特别是大型软件系统的开发，并支持面向规格说明、面向过程、面向对象等多种软件开发方法，是一种具有广阔前景的模型

7. 喷泉模型

- a. 定义：一种以用户需求为动力，以对象为驱动力的模型，主要用于描述面向对象的软件开发过程。
- b. 优点：喷泉模型不像瀑布模型那样，需要分析活动结束后才开始设计活动，设计活动结束后才开始编码活动。该模型的各个阶段没有明显的界限，开发人员可以同步进行开发。其优点是可以提高软件项目的开发效率，节省开发时间，适应于面向对象的软件开发过程。
- c. 缺点：由于喷泉模型在各个开发阶段是重叠的，在开发过程中需要大量的开发人员，因此不利于项目的管理。此外，这种模型要求严格管理文档，使得审核的难度加大，尤其是面对可能随时加入各种信息、需求与资料的情况。

8. 基于构件的模型

a. 四个阶段

- 需求
- 组件分析：根据需求规格搜索可满足该需求的组件。通常情况下，没有完全匹配的情况，因而组件通常需要加以修改。
- 系统设计：与其他模型的系统设计有所不同，因为该模型是基于重用的。设计者必须考虑到重用的概念，但遗憾的是，如果没有可重用的组件，还要设计新的软件。
- 开发和集成：在这个阶段，组件集成到系统中。

b. 优点：组建的重用，降低了成本和风险，节约了时间

c. 缺点

- 模型复杂
- 导致需求的折衷，进而导致系统不能完全符合需求
- 无法完全控制所开发系统的演化
- 项目划分的好坏直接影响项目结果的好坏

9. 敏捷开发模型

- a. 思想：将大项目分为相互联系但又可以独立运行的小项目，分别完成，在此过程中，软件处于可使用状态
 - b. 优点：紧密协作，面对面沟通
 - c. 缺点：文档少
 - d. 特点
 - 有效响应变化
 - 利益相关者间的有效沟通
 - 客户加入开发团队
 - 高度自主的项目团队
 - 快速交付的软件增量
 - e. 适用场景：小型项目
10. 软件过程模型时间表
- o 增量模型：1968
 - o 瀑布模型：1970
 - o 螺旋模型：1988
 - o 原型模型：1989
 - o 敏捷模型：2001
11. 如何选择过程模型
- o 软件开发模型是不断发展的
 - o 各种软件开发模型各有优缺点
 - o 选用时不必拘泥于某种模型
 - o 可组合多种模型
 - o 也可根据实际创建新的模型
- 参考原则
- o 在前期需求明确的情况下，尽量采用瀑布模型或改进的瀑布模型
 - o 在用户无系统使用经验，需求分析人员技能不足的情况下一定借助原型
 - o 在不确定因素很多，很多东西前面无法计划的情况下尽量采用增量迭代和螺旋模型
 - o 在需求不稳定的情况下尽量采用增量迭代模型
 - o 在资金和成本无法一次到位的情况下可采用增量模型，软件产品分多个版本进行发布
 - o 对于完成多个独立功能开发可以在需求分析阶段就进行功能并行，但每个功能内部都应该遵循瀑布模型
 - o 对全新系统的开发必须在总体设计完成后再开始增量或并行
 - o 对于编码人员经验较少的情况下建议不要采用敏捷或迭代等生命周期模型
 - o 增量、迭代和原型可以综合使用，但每一次增量或迭代都必须有明确的交付和出口原则
 - o 过程与产品：以产品为中心还是以过程为中心？过程

名称	优点	缺点	适用条件
瀑布模型	1. 它提供了一个模板，使得软件分析、设计、编码、测试、维护工作都可以在该模板的指导下进行，避免了软件开发过程中随意的状态 2. 对于需求确定、变动较少的项目，线性顺序模型仍然是一种可采用的过程模型。采用这种模型曾经成功进行过许多大型软件工程的开发	1. 各个阶段的划分完全固定，阶段之间产生大量的文档，极大增加了软件开发的工作量 2. 由于开发过程是线性的，用户只能等到开发最后阶段才能看到结果，极大地增加了风险。 3. 前期的错误可能到后期的测试阶段才能发现，进而带来更严重的后果	瀑布模型适用于系统需求明确、技术成熟、工程管理较严格的场合。
增	1. 增量包概念的引入，以及不需要	每个增量必须提供一些系统	适用于软件开发中

量 模 型	<p>提供完整的需求。只要有一个增量包出现，开发就可以进行。</p> <p>2. 在项目开始不需要投入太多的人力资源</p> <p>3. 增量可以有效地管理技术风险</p>	<p>功能，这使得开发者很难根据客户需求给出大小合适的增量</p>	<p>需求可能变化，具有较大风险，希望可以尽早进入市场的软件工程。</p>
原 型 模 型	<p>1. 克服了瀑布模型的缺点，减少了由需求不明确带来的风险，出现需求变更，能迅速做出调整</p> <p>2. 开发工具先进，开发效率高，成本低</p>	<p>1. 设计者在质量和原型间有所折衷</p> <p>2. 客户意识不到一些质量问题</p>	<p>客户定义一个总体目标集，但他们并不知道系统具体的输入输出，或者开发者不确定算法的效率、软件与操作系统是否兼容以及客户与计算机交互的方法</p>
螺 旋 模 型	<p>1. 支持需求的动态变化</p> <p>2. 原型可以看作是一个形式的、可执行的需求规格说明，易于被开发人员和用户共同理解。原型也可以作为继续开发的基础，为用户参与所有关键决策提供了便利</p> <p>3. 螺旋模型特别强调原型的可修改性和可扩充性，原型的进化贯穿于整个软件开发生命周期，这将有助于提高目标软件的适应能力</p> <p>4. 模型为管理人员及时调整管理决策提供了方便，进而降低了开发风险</p>	<p>1. 如果每次迭代的效率不高，致使迭代次数过多，将会增加成本并推迟提交时间。</p> <p>2. 使用该模型需要有相当丰富的风险评估经验和专门知识，要求开发队伍水平较高</p>	<p>支持需求不明确，特别是大型软件系统的开发，并支持面向规格说明、面向过程、面向对象等多种软件开发方法，是一种具有广阔前景的模型</p>
喷 泉 模 型	<p>喷泉模型不像瀑布模型那样，需要分析活动后才开始设计活动，设计活动结束后才开始编码活动。该模型各个阶段没有明显的界限，开发人员可以同步进行开发。其优点是可以提高软件项目的开发效率，节省开发时间，适用于面向对象的软件开发过程</p>	<p>由于喷泉模型在各个阶段是重叠的，在开发过程中需要大量的人员，因此不利于项目的管理。此外，这种模型要求严格管理文档，使得审核的难度加大，尤其是面对可能随时假如各种信息，需求与资料的情况</p>	<p>适用于面向对象的软件开发过程</p>