

软件工程

2.过程和生命周期建模

1.什么是软件？

2.什么是软件危机，其内容主要是指什么？

3.什么是软件工程？

4.软件工程的目标及其组成部分。

5.软件开发方法的定义

6.软件工程方法三要素

7.好的软件的一些主要衡量指标

是一系列按照特定顺序组织的计算机数据和指令的集合，包括程序、数据和文档。

软件危机是指落后的软件生产方式无法满足迅速增长的计算机软件需求，从而导致软件开发与维护过程中出现一系列严重问题的现象。

概括地说，软件工程是指导计算机软件开发和维护的工程学科。采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，以经济地开发出高质量的软件并有效地维护它，这就是软件工程。

目标：软件工程的目标是运用先进的软件开发和管理方法来提高软件的质量和生产率，也就是要以较短的周期、较低的成本生产出高质量的软件产品，并最终实现软件的工业化生产。

通常将软件开发生命周期全过程中使用的一整套技术方法的集合称为方法学，也称为范型。

目前使用最广泛的软件方法学：传统方法学（结构化方法学），面向对象方法学。

软件工程方法为软件开发提供了“如何做”的技术；

软件工具为软件工程方法提供了自动的或半自动的软件支撑环境；

过程是为了获得高质量的软件所需要完成的一系列任务框架，它规定了完成各项任务的工作步骤；

- 功能性是指软件所实现的功能达到它的设计规范和满足用户需求的程度，其质量子特征包括适合性、正确性、安全保密性；
- 可靠性是指在规定的时间和条件下，软件能够正常维持其工作的能力，其质量子特征包括成熟性、恢复性、容错性；
- 易用性是指为了使用该软件所需要的能力，其质量子特征包括易理解性、易学习性、易操作性；
- 效率是指规定的条件下用软件实现某种功能所需要的计算机资源的有效性，其质量子特征包括时间特性、资源特性；
- 可维护性是指当环境改变或软件运行发生故障时，为使其恢复正常运行所做努力的程度，其质量子特征包括易分析性、易修改性、易测试性；
- 可移植性是指软件从某一环境转移到另一环境时所做努力的程度，其质量子特征包括适应性、易替换性。

概念：软件也有一个孕育、诞生、成长、成熟和衰亡的生存过程，我们称这个过程为软件生命周期或软件生存期。

分为软件定义、软件开发、运行维护三个阶段

软件定义：解决“做什么”问题（问题定义、可行性研究、需求分析三个阶段）

- 问题定义
 - 关键问题是：“到底要解决什么问题？”
 - 提交内容：关于问题性质、工程目标和工程规模的书面报告
 - 关键问题：“上一个阶段所确定的问题是否有行得通的解决办法”
- 可行性研究
 - 提交内容：可行性研究报告，即从技术、经济和社会因素等方面研究各方案的可行性
- 需求分析和定义
 - 任务
 - 对用户提出的要求进行分析并给出详细的定义
 - 准确地回答“目标系统必须做什么”这个问题。也就是对目标系统提出完整、准确、清晰、具体的要求
 - 提交内容
 - 编写软件需求说明书或系统功能说明书及初步的系统用户手册

概括地说，软件工程是指导计算机软件开发和维护的工程学科。采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，以经济地开发出高质量的软件并有效地维护它，这就是软件工程。

确定总目标和可行性；
导出策略和系统功能；
估计资源或成本；
制定工程进度表。

软件开发：解决“如何做”问题

- 任务：具体设计和实现前一个时期即软件定义时期定义的软件
- 分为概要设计、详细设计、编码和测试四阶段。前两个阶段又称为系统设计，后两个阶段称为系统实现
- 概要设计
 - 概括地回答“怎样实现目标系统？”
 - 设计程序的体系结构，也就是确定程序由哪些模块组成以及模块间的关系
 - 提交的文档是概要设计说明书
- 详细设计
 - 回答“应该怎样具体地实现这个系统”
 - 详细地设计每个模块，确定实现模块功能所需要的算法和数据结构
 - 提交的文档是软件的设计说明书
- 编码
 - 写出正确的、容易理解、容易维护的程序模块
 - 提交的文档为源程序、详尽的程序说明和单元测试报告
- 测试
 - 包括集成测试和系统测试
 - 通过各种类型的测试(及相应的调试)使软件达到预定的要求
 - 提交的文档为测试计划、详细测试方案以及实际测试结果等
- 运行维护：使软件持久地满足用户的需求
 - 适应性维护：修改软件以适应环境的变化
 - 完善性维护：根据用户的要求改进或扩充软件，使它更完善
 - 预防性维护：即修改软件为将来的维护活动预先做准备

开发文档

概要设计和详细设计阶段

用户手册

1.什么是软件生命周期？主要分为哪些阶段？各个阶段的主要任务及产生的主要制品？

2.开发过程的典型文档

2.需求分析的定义

- 软件需求规格说明书：描述将要开发的软件做什么。
- 项目计划：描述将要完成的任务及其顺序，并估计所需要的时间及工作量。
- 软件测试计划：描述如何测试软件，使之确保软件应实现规定的功能，并达到预期的性能。
- 软件设计说明书：描述软件的结构，包括概要设计以及详细设计。
- 用户手册：描述如何使用软件。

- 对用户提出的要求进行分析并给出详细的定义
- 准确地回答“目标系统必须做什么”这个问题。也就是对目标系统提出完整、准确、清晰、具体的要求
- 编写软件需求说明书或系统功能说明书及初步的系统用户手册
- 提交管理机构评审

特点

- 阶段间具有顺序性和依赖性（包含两方面）
 - 必须等前一阶段的工作完成之后，才能开始后一阶段的工作；
 - 前一阶段的输出文档就是后一阶段的输入文档；
- 推迟实现的观点（两方面）
 - 瀑布模型在编码之前设置了系统分析和系统设计的各个阶段，分析与设计阶段的基本任务规定，在这两个阶段主要考虑目标系统的逻辑模型，不涉及软件的物理实现。
 - 清楚地区分逻辑设计与物理设计，尽可能推迟程序的物理实现，是按照瀑布模型开发软件的一条重要的指导思想。
- 质量保证的观点（两方面）
 - 每个阶段都必须完成规定的文档，没有交出合格的文档就是没有完成该阶段的任务。
 - 每个阶段结束前都要对所完成的文档进行评审，以便尽早发现问题，改正错误。

优点

- 可强迫开发人员采用规范化的方法。
- 严格地规定了每个阶段必须提交的文档。
- 要求每个阶段交出的所有产品都必须是经过验证的。

缺点

- 瀑布模型只适用于项目开始时需求已确定的情况
- 特点：快速原型是快速建立起来的可以在计算机上运行的程序，它所能完成的功能往往是最终产品能完成的功能的一个子集

快速原型模型

- 1) 有助于满足用户的真实需求。
- 2) 原型系统已经通过用户的交互而得到验证，据此产生的规格说明文档能够正确地描述用户需求。
- 3) 软件产品的开发基本上是按线性顺序进行。
- 4) 因为规格说明文档正确地描述了用户需求，因此，在开发过程的后续阶段不会因为发现规格说明文档的错误而进行较大的返工。
- 5) 开发人员通过建立原型系统已经学到了许多东西，因此，在设计和编码阶段发生错误的可能性也比较小，这自然减少了在后续阶段需要改正前面阶段所犯错误的可能性。
- 6) 快速原型的突出特点是“快速”，开发人员应该尽可能地建造出原型系统，以加速软件开发过程，节约软件开发成本。

使用增量模型开发软件时，把软件产品作为一系列的增量构件来设计、编码、集成和测试。

增量模型（建增模型）

- 每个构件由多个相互作用的模块构成，并且能够完成特定的功能。
- 优点
 - 1) 能在较短时间内向用户提交可完成一些有用的工作产品，即从第1个构件交付之日起，用户就能做一些有用的工作。
 - 2) 逐步增加产品的功能可以使用户有较充裕的时间学习和适应新产品，从而减少一个全新的软件可能给用户组织带来的冲击。
 - 3) 项目失败的风险较低，虽然在某些增量构件中可能遇到一些问题，但其他增量构件将能够成功地交付给用户。
 - 4) 优先级最高的服务首先交付，然后再将其他增量构件逐次集成进来，因此，最重要的系统服务将接受最多的测试。
- 需要注意
 - 1) 在把每个新的增量构件集成到现有软件体系结构中时，必须不破坏原来已经开发出的产品。
 - 2) 软件体系结构必须是开放的，即向现有产品中加入新构件的过程必须简单、方便。

螺旋模型（加入风险分析）

- 基本思想：使用原型及其他方法来尽量降低风险
- 优点
 - 对可选方案和约束条件的强调有利于已有软件的重用，也有助于把软件质量作为软件开发的一个重要目标。
 - 减少了过多测试或测试不足所带来的风险。
 - 在螺旋模型中维护只是模型的另一个周期，因而在维护和开发之间并没有本质区别。
- 缺点
 - 螺旋模型是风险驱动的，因此要求软件开发人员必须具有丰富的风险评估经验和这方面的专门知识，否则将出现真正的风险：当项目实际上正在走向灾难时，开发人员可能还以为一切正常

喷泉模型（面向对象）

原型法是系统分析设计人员与用户紧密合作，在短时间内定义用户的基本需求，开发出一个功能不十分完善、实验性的系统基本框架（称为系统原型），先运行这个原型，再不断评价和改进这个原型，直至系统完善和满意为止。

4.原型法的特点及分类

- 分类
 - 探索型原型、实验型原型、演化型原型

5.敏捷开发方法和极限编程的特点。

- 敏捷开发方法
 - 客户应该和开发团队在一起密切工作
 - 通过高度迭代和增量式的软件开发过程响应变化
 - 通过频繁地提供可以工作的软件来搜集人们对产品的反馈
- 极限编程
 - 简单的分析设计
 - 频繁的客户交流
 - 增量式开发和连续的测试

3.获取需求

1.了解需求的重要性及需求分析阶段的目标及主要产物。

2.需求工程包括哪些方面？

- 重要性
 - 软件需求分析是软件生存期中重要的一步，也是决定性的一步
 - 只有通过软件的需求分析活动才能把软件功能和性能的总体概念描述为具体的软件需求规格说明，从而奠定软件开发的基础
- 目标（任务）
 - 发现和分析问题，并分析问题的原因/结果关系。
 - 与用户进行各种方式的交流，并使用调查研究方法收集信息。
 - 按照三个成分观察问题的不同侧面：即数据、过程和接口。
 - 将获取的需求文档化，形式有用例、决策表、需求表等。
- 产物
 - 软件需求规格说明书
 - 系统功能说明书
 - 初步的用户手册
- 需求开发（获取、分析、定义、验证）
 - 1) 获取需求
 - 2) 分析需求
 - 3) 定义需求
 - 4) 验证需求
- 需求管理
 - 变更控制
 - 需求跟踪
 - 需求状态跟踪
 - 需求文档版本控制
- 功能需求
 - 性能需求、资源和环境需求、可靠性需求、安全保密要求、用户界面需求
- 非功能需求
 - 成本消耗与开发进度需求、预先估计的可扩展性需求。

3.需求的类型：功能需求、非功能需求或质量需求、设计约束、过程约束

4.两种需求文档：需求定义文档和需求规格说明书。

5.需求规格说明书的主要内容

- 引言：编写目的、背景、定义、参考资料
- 任务概述：目标、用户特点、假设与约束
- 需求规定：对功能的规定、对性能的规定、输入输出要求、数据管理能力要求、故障处理要求
- 运行环境规定：设备、支持软件、接口、控制
- 需求分析

6.常用的需求建模表示方法：ER图、事件跟踪、状态机、Petri网、数据流图、用例图和原型法。