

专 业 综 述

指导教师： 李晓宇 职称： 副教授

指导教师(校外)： 张创伟 职称： 研发经理

学生姓名： 李伟 学号： 20162430211

专 业： 软件工程

院（系）： 信息工程学院

完成时间： 2020 年 6 月 7 日

2020 年 6 月 7 日

软件工程综述

摘要: 本文简单介绍了软件工程。分别从软件工程的定义、发展历程、危机及解决途径、软件过程的基本原理、软件工程包含的领域和生命等方面介绍。从而对软件工程的基本内容有了初步的了解。

1. 引言

软件工程 (**Software Engineering** , 简称为 **SE**) 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。它涉及到程序设计语言, 数据库, 软件开发工具, 系统平台, 标准, 设计模式等方面。在现代社会中, 软件应用于多个方面。典型的软件比如有电子邮件, 嵌入式系统, 人机界面, 办公套件, 操作系统, 编译器, 数据库, 游戏等。同时, 各个行业几乎都有计算机软件的应用, 比如工业, 农业, 银行, 航空, 政府部门等。这些应用促进了经济和社会的发展, 使得人们的工作更加高效, 同时提高了生活质量。

2. 系统介绍

2.1 软件工程的定义

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科技方法结合起来, 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。

下面给出软件工程的几个定义:

1983 年 IEEE (美国电气和电子工程师协会) 给软件工程下的定义是: “软件工程是开发、运行、维护和修复软件的系统方法。” 这个定义相当概括, 它主要强调软件工程是系统方法而不是某种神秘的个人技巧。

Fairly 认为: “软件工程学是为了在成本限额以内按时完成开发和修改软件产品所需要的系统生产和维护技术及管理学科。” 这个定义明确指出了软件工程的目标是在成本限额内按时完成开发和修改软件的工作, 同时也指出了软件工程包含技术和管理两方面的内容。

Fritz Bauer 给出了下述定义: “软件工程是为了经济地获得可靠的且能在实际机器上有效地运行的软件, 而建立和使用的完善的工

程化原则。”这个定义不仅指出软件工程的目的是经济地开发出高质量的软件，而且强调了软件工程是一门学科，它应该建立并使用完善的工程化原则。

1993 年 IEEE 进一步给出了一个更全面的定义。

软件工程是：①把系统化的、规范的、可度量的途径应用于软件开发、运行和维护的过程，也就是把工程化应用于软件中；②研究①中提到的途径。

2.2 软件发展及危机

2.2.1 软件发展的历程

程序设计阶段 计算机发展的早期阶段（20 世纪 50 年代初期~20 世纪 60 年代中期）。在这个阶段硬件已经同用花，而软件的生产却是个体化。

程序系统阶段 计算机系统发展的第二阶段（20 世纪 60 年代中期~20 世纪 70 年代末期）。引入了人机交互的新概念。出现了实时系统和第一代数据库管理系统。

软件工程阶段 计算机系统发展的第三阶段（始于 20 世纪 70 年代中期）。以软件的产品化、系列化、工程化、标准化为特征的软件产业发展起来，打破了软件生产的个体化特征，有了可以遵循的软件工程化得设计原则、方法和标准。

第四阶段 有复杂操作系统控制的强大的桌面机、广域网络和局域网络，配以先进的软件应用已成为标准。从技术的软件刚才阶段过渡到社会信息化得计算机系统阶段。

随着第四阶段的进展，一些新技术开始涌现。

2.2.2 软件危机

(1)由于缺乏软件开发经验和科学的理论指导，开发者不能准确地估计软件开发的成本和进度。所制定的成本规划可能大大低于实际的成本，实际的开发进度也可能要比计划的慢，为了赶进度往往就会牺牲软件的质量。

(2)由于缺乏使开发人员与用户进行交流的有效机制，开发人员常会犯闭门造车的错误。从而会造成用户对已完成的软件系统不满意。

(3)由于测试工作不够充分，又没有好的软件质量保证技术，导致

提交给用户的软件质量不高。

(4)软件的可维护性差,程序的错误难以改正。程序的可移植性差,很难适应不同的运行环境。软件的可重用性差,大量的软件人员在重复开发。

(5)开发过程缺乏标准和规范的指导,各个开发组织都有自己的开发方法,开发组织之间进行工作交接的流程也不规范。

(6)软件开发生产率的提高速度难以满足对软件需求的增长速度,软件产品供不应求。

(7)由于软件开发和维护方法不当,使得软件成本居高不下,过高的成本已严重制约了软件的开发。

造成软件危机的原因很多,归纳起来主要有两个方面:一是与软件本身的特点有关,二是与软件开发与维护方法不当有关。

2.2.3 解决软件危机的途径

(1)采用工程化方法来开发和维护软件。软件开发不应只是个体化的劳动,而应该是由组织良好、管理严密、各类人员共同配合完成的一个工程项目,因此应该注意吸收和借鉴从事其他工程项目的行之有效的科学原理和方法。

(2)采用先进的技术、方法、工具开发和软件设计。即采用先进的管理技术、规范的开发方法和模型、各种提高开发效率的软件工具等。

(3)采用必要的组织管理措施。软件工是正式在解决软件危机问题的过程中形成的一门综合技术与管理两个方面的新兴学科,并逐渐成为指导计算机开发、维护、管理的理论依据。

2.3 软件过程的基本原理

2.3.1 软件的生命周期

一个软件产品从构想到不再使用,期间会经历若干阶段,我们称其为软件的生命周期(life cycle)。生命周期中的 3 个主要阶段是:开发阶段(development),使用阶段(use) 和维护阶段(maintenance)。通常情况是:客户提出需要解决的问题,软件开发者就构思并开发相应的软件,并藉此获得酬劳。新的软件是在开发阶段建立的。

软件开发完毕之后就交付用户使用。用户在使用过程中,通常都

会发现各种问题（错误）及提出各种修改建议。这些信息都会反馈给开发者，这就进入了维护阶段。在软件维护阶段中，软件中的错误被修改（标识），功能被增强。如果需要进行较大的修改，通常会开发一个新版本的软件并发布和使用。当一个软件的维护费用过于昂贵时，开发者就考虑不再使用该软件，也不再发布新的版本。软件开发阶段是生命周期中的第一个阶段，也是最重要的阶段。如果一个软件开发得好，后续的维护将十分容易，相应的也就节约维护成本。

2.3.2 软件开发阶段

软件工程师通常将软件的开发阶段分为下面 4 个子阶段：

- 分析阶段
- 设计阶段
- 实现阶段
- 测试和调试阶段

1. 分析阶段

分析问题是第一步也是最为重要的一部。在这一步中。您需要做以下事情：

- 全面理解所要解决的问题
- 进行需求分析（ **requirement analysis** ）：理解问题需求，包括程序是否需要和用户进行交互，是否操纵数据，是否有输出结果以及输出结果的格式等等。举一个例子，如果您要编写一个自动取款机（ **ATM** ）的程序。在分析阶段，您要给出该 A T M 机可以进行的操作，如：取款，存款，转账及余额查询等等。您会和使用该 A T M 机的客户进行讨论，理解他们的需求，增加必要的操作，以便是您的程序是用户友好的。
- 如果程序需要对数据进行操作，开发人员必须了解数据类型及它们的表示方法。这时候可能会接触一些样本数据。如果程序有输出信息，必须确定它们所生成的结果及输出格式等。
- 如果需要解决的问题过于复杂，可以把它分解为多个子问题，在对每个子问题做相应的需求分析。

2. 设计阶段

当您仔细分析完问题后，就需要设计相应的算法去解决问题。如果已经将问题分解为若干子问题，那么对每个子问题都需要设计相应的算法。

算法（ **algorithm** ）：在有限时间内获得问题解决方案的逐步求解过程。

○ 1 结构化设计方法

将一个问题分解为若干个子问题的方法叫做结构化设计方法（ **structured design** ）。结构化设计方法又叫做自顶向下的设计方法（ **top-down design** ）、逐步求精方法（ **stepwise refinement** ）和模块化程序设计方法（ **modular programming** ）。在结构化设计方法中，问题被分解为若干子问题，然后分别对每个子问题进行分析和求解。所有子问题的解合并起来就是原始问题的解。使用结构化设计方法进行编程就叫做结构化程序设计（ **structure programming** ）。

○ 2 面向对象设计方法

在面向对象设计方法（ **object-oriented** ， **OOD** ）中，求解问题的首要步骤是识别称为“对象”（ **object** ）的组件（它是运用该方法求解问题的基础）和确定对象之间如何进行交互。比如，我们要为一家录像出租店编写一个可以使其业务自动化的程序。在这个问题中，我们可以确定两个主要对象，即：录像带对象和客户对象。明确系统中的对象之后，下一步就是为每个对象确定有关数据和数据上进行的相关操作。对于录像带对象而言，数据可能包括影片名称，演员名单，制片人，出版公司，库存副本数等等。在录像带对象可以进行的相关操作包括查询影片名，每租出一盘录像带就将库存副本数减一，客户每归还一盘录像带就将库存数加一。

由上文可知：对象包括数据和在数据上执行的操作。对象可以看作数据和其上操作的统一体。使用面向对象方法编程，最终的程序是交互对象的集合。实现面向对象设计方法的编程语言叫做面向对象程序设计（ **object-oriented programming** ， **OOP** ）语言。

面向对象设计方法的 3 个基本原则：

- 封装性（ **encapsulation** ）：将数据和操作集成在一个单元（对象）中的能力。
- 继承性（ **inheritance** ）：从已有数据类型中派生新数据类型的能力。
- 多态性（ **polymorphism** ）：使用相同表达形式来实现不同操作的能力。

在 Java 中，封装性是通过叫做类（ **class** ）的数据类型实现的。在面向对象的设计方法中，我们要确定需要哪些类，它们的数据成员

和成员方法。我们还需要描述个各类之间如何进行交互。

4. 实现阶段

在实现阶段，您将编写和编译程序代码，以实现在设计阶段分析得到的类和函数。最终的程序通常由几个函数组成，它们分别完成不同的任务。有些函数是主程序的一部分，其他的则是在对象上完成的操作。显然函数之间存在交互关系，从而能够利用彼此的函数功能。要使用一个函数，用户只需要知道该函数的功能和用法即可，而不必关心该函数的实现的具体细节。

5. 测试和调试

测试（ **testing** ）这个术语表示检测程序的正确性，即检查程序是不是完成了需要完成的工作。而调试（ **debugging** ）一词指，如果程序存在错误，如何找到并修改错误。在每写完一个函数或算法后，接下来应该验证它是否正确工作。在复杂的大型程序中，错误是一定存在的。为了提高程序的可靠性，必须在交付用户前发现并修改其中的错误。您可以使用某些方法，通常是数学方法来证明程序的正确性。然而，对于大型的程序来说，单单使用证明方法是不行的，因为在证明过程中就有可能引入错误。所以，我们必须使用测试的方法考察程序的质量。通过让程序运行一些特定的例子，即测试用例（ **test case** ）来找出程序中的问题。

测试用例的组成部分包括：一组输入数据、用户操作、初始条件和期望的结果等。由于测试用例将被多次使用，所以需要正确地记录它们。通常，一个程序需要对大量的数据进行操作，虽然理论上可以将所有的输入情况在测试用例中体现，但在实际工作中显然是不现实的。例如，一个程序需要对整数进行处理，显然，为每个整数做一个测试用例是不可能的。我们可以将测试用例分类，即分为等价类（ **equivalence category** ）。所谓等价类是指：在这个分类中的所有输入值将对应相同的输出值。有比如说，有一个函数的输入为整数，并且如果该整数为非负则返回真（ **true** ），否则返回假（ **false** ）。那么我们可以做两个等价类，一个为负数集合，另一个为非负数集合。

测试有两类方法，即：黑盒测试（ **black-box testing** ）和白盒测试（ **White-box testing** ）。使用黑盒测试方法时，您不需要知道算法或函数的内部实现，只需要知道程序的功能即可黑盒测试是基于输入输出的方法。它的测试用例通过创建等价类来选取。如果程序对于等价类中的某个输入的输出结果是正确的话，那么就认为对应该等价类

中其他输入也会输出同样的正确结果。

软件工程的目标：提高软件的质量与生产率，最终实现软件的工业化生产。

2.3.3 原理

1. 用分阶段的生命周期计划严格管理
2. 坚持进行阶段评审
3. 实行严格的产品控制
4. 采用现代化程序设计技术
5. 结果应能清楚地审查
6. 开发小组的人员应该少而精
7. 承认不断改进软件工程实践的必要性

2.4 软件工程包含的领域

2.4.1 软件需求

用户解决问题或达到目标所需条件或权能；系统或系统部件要满足合同、标准、规范或其它正式规定文档所需具有的条件或权能；一种反映上面所述条件或权能的文档说明。

2.4.2 软件设计

把许多事物和问题抽象起来，并且抽象它们不同的层次和角度。是一个过程和这个过程的结果，此过程对一个系统或组件定义架构（体系结构）、组件、接口以及其他特征。其在软件开发中起着重要的作用。

2.4.3 软件构建

指的是如何创建产生软件的详细步骤，包括编码、验证、单元测试、集成测试和调试。

2.4.4 软件测试

利用测试工具按照测试方案和流程对产品进行功能和性能测试，甚至根据需要编写不同的测试工具，设计和维护测试系统，对测试方

案可能出现的问题进行分析和评估。执行测试用例后，需要跟踪故障，以确保开发的产品适合需求。

2.4.5 软件维护

软件开发的工作的结果就是交付一个满足用户需求的软件产品。软件产品一旦投入应用，产品的缺陷就会逐渐的暴露出来，运行的环境会逐渐发生变化，新的用户也会不断地浮出水面。软件维护就是要针对这些问题而对软件产品进行相应的修改或演化，从而真正的修改错误，改善性能或其他特征。

2.4.6 软件配置管理

一项跟踪和控制软件变更的活动

2.4.7 软件工程管理

是软件的开发和维护的管理活动，为了达到系统的、遵守规程的和可量化的目的，它包括计划、协调、度量、监控、控制和报表。

2.4.8 软件工程过程

将软件工程的方法和工具综合起来，以达到合理、及时地进行计算机软件开发的目的。软件工程过程应确定方法使用的顺序、要求交付的文档资料、为保证质量和适应变化所需要的管理、软件开发各个阶段完成的任务。

3. 结论

本文给出了关于软件工程的综合论述，阐明了软件工程的一些基本原理，对软件工程的基本情况进行了介绍，加深了我对软件件工程的理解。