

CSE1 软件开发概述

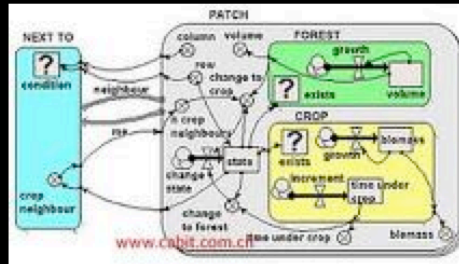
没有银弹

- 软件的开发是一种高度复杂的智力活动。
- 1960's年代，人们提出了“软件危机”的说法，来描述软件开发的困难，并提出了“软件工程”的概念。
- Brooks在1986年“没有银弹”一文中提到软件开发具有复杂性、不一致性、可变性和不可见性，说明了软件开发是一项困难的任务，不存在特效药。

软件开发

- 使用软件解决我们现实世界中的问题。
- 现代计算机只能执行机器世界中的二进制代码。
- 通常，我们会先用自然语言来描述现实世界的问题；然后通过智力活动对该问题进行思考，为其设计计算机可以完成的解决方案；最后用具体的程序设计语言来实现该解决方案，并通过翻译程序翻译成二进制代码，在计算机上执行以解决该问题。

概念结构



高级语言

中级语言

低级语言

编译器

人



问题

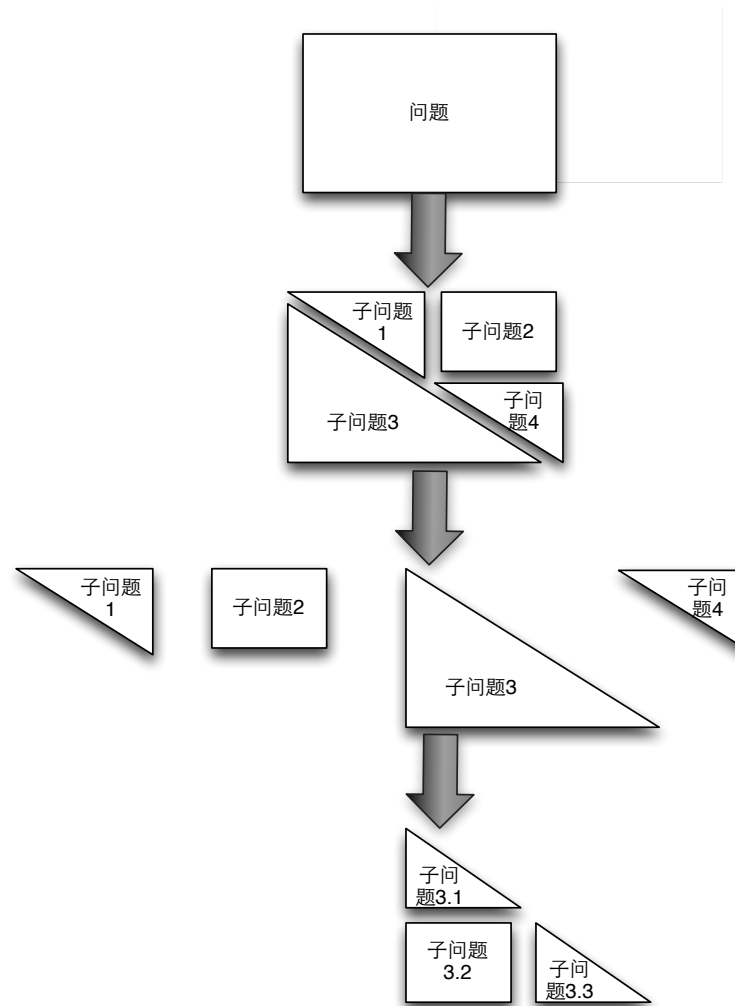
- 人们通过“自然语言”来描述遇到的问题。
- 计算机无法理解自然语言。
- 自然语言有二义性。

解决方案

- “一个相互牵制关联的概念结构，是软件实体必不可少的部分，它包括：数据集合、数据条目之间的关系、算法、功能调用等等。”[Brooks, 1986]。软件开发通过智力活动，根据计算机的运行原理，所具有的计算能力，来设计针对某个特定问题的解决方案。

解决复杂性

- 复杂性是软件开发面临的主要困难之一。
- 问题分解。



算法

- 在确定的解决方案中，我们会形成一个相互牵制关联的概念结构，其中的算法是指完成一个任务所需要的具体步骤和方法。
- 以下是Donald Knuth在他的著作《The Art of Computer Programming》里对算法下的定义：
 - 输入：一个算法必须有至少零个输入量。
 - 输出：一个算法应有至少一个输出量，输出量是算法计算的结果。
 - 明确性：算法的描述必须无歧义，以保证算法的实际执行结果是精确地符合要求或期望，通常要求实际运行结果是确定的。
 - 有限性：依据图灵的定义，一个算法是能够被任何图灵完备系统模拟的一串运算，而图灵机只有有限个状态、有限个输入符号和有限个转移函数（指令）。而一些定义更规定算法必须在有限个步骤内完成任务。
 - 有效性：又称可行性。能够实现，算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现。

编程

- 确定了解决方案以后，接下来的工作就是使用一种程序设计语言，将解决方案转换成程序。
- 程序设计语言是根据计算机运行原理设计的语言，所以它与自然语言的差异较大，是计算机指令表达的一种方式，不存在歧义性。

机器语言与汇编语言

- 机器语言（Machine Language）是机器的自然语言，它是唯一一种计算机能直接理解并执行的语言。
- 对机器语言进行上层抽象，形成了汇编语言，提高了可读性与可移植性，使得程序变得更加容易修改和调试。

面向问题的语言

- 高级语言使用的是易于理解的符号和英文单词组成的语句，接近自然语言，独立于计算机硬件系统，每条语句都有相当于若干条低级语言语句的功能。
- 如目前流行的Java，C++，Python，PHP，C#，（Visual）Basic，Objective-C，JavaScript等

TIOBE Programming Community Index for February 2013

Position Feb 2013	Position Feb 2012	Delta in Position	Programming Language	Ratings Feb 2013	Delta Feb 2012	Status
1	1	=	Java	18.387%	+1.34%	A
2	2	=	C	17.080%	+0.56%	A
3	5	↑↑	Objective-C	9.803%	+2.74%	A
4	4	=	C++	8.758%	+0.91%	A
5	3	↓↓	C#	6.680%	-1.97%	A
6	6	=	PHP	5.074%	-0.57%	A
7	8	↑	Python	4.949%	+1.80%	A
8	7	↓	(Visual) Basic	4.648%	+0.33%	A
9	9	=	Perl	2.252%	-0.68%	A
10	12	↑↑	Ruby	1.752%	+0.19%	A
11	10	↓	JavaScript	1.423%	-1.04%	A
12	16	↑↑↑↑	Visual Basic .NET	1.007%	+0.21%	A
13	13	=	Lisp	0.943%	+0.04%	A
14	15	↑	Pascal	0.932%	+0.12%	A
15	11	↓↓↓↓	Delphi/Object Pascal	0.886%	-1.08%	A
16	14	↓↓	Transact-SQL	0.773%	-0.07%	A--
17	75	↑↑↑↑↑↑↑↑	Bash	0.741%	+0.61%	A--
18	26	↑↑↑↑↑↑↑↑	MATLAB	0.648%	+0.15%	B
19	24	↑↑↑↑↑	Assembly	0.640%	+0.12%	B
20	19	↓	Ada	0.631%	0.00%	B

选择Java作为本课程学习语言 (技术)

-来自知乎

- 1、至今为止，java是大量实际得到应用的语言中，可读性最强，最利于阅读和理解，语法最严谨和规范的语言（当然，这也可以理解为语法繁琐的另一种表述）之一。
 - java是目前最佳的算法及数据结构教学语言
 - java是传统的软件开发过程（生命周期管理）的最佳实践语言，即在传统的“需求-设计-代码-测试”这样一个过程中，java是最能够最大程度贯彻和实践软件工程学的理论的。
 - java的白盒测试方面的表现非常出色和易行
 - java在代码评审，缺陷管理，开发规范约束，大团队的协同开发方面，有着无可争辩的突出优势

- 2. java有着目前为止，最丰富，最强大的IDE开发环境，这是历史原因形成的，包括商业因素在内
- 3. java有着现有所有语言中，拥有最长的产品线，适用性是最广的语言之一。从前端的app应用，到嵌入式，到web，到服务器应用
- 4. java拥有所有语言中最丰富的类库和代码资源
- 5. 至今为止，在被广泛应用的编程语言中，java依然是开发效率最高的语言。之所以使用编程语言这个词，是为了和脚本语言，以及4GL开发工具区隔开来。这些语言和编程语言相比，其适用性都有很大局限，而且后者（4GL）和OS平台紧耦合关联。事实上，近10多年来，脚本语言领域突飞猛进（有赖于web应用提供了广阔的舞台），而编程语言基本上自java之后就没有太多大的新鲜事了。

商业层面

- 1. java是目前为止唯一的，在商业和开源领域都得到大力推广，推荐和使用的语言，其背后的推力是任何语言所不能比拟的。
- 2. java的诞生和发展，赶上了千载难逢的好机遇，历史造就了java。

高级语言

- 优点
 - ①高级语言具有易学性
 - ②高级语言具有易读性。
 - ③高级语言程序易移植。
 - ④高级语言为程序员带来了便捷。
- 缺点
 - ①高级语言“翻译”后，目标程序冗长。
 - ②高级语言“翻译”后，目标程序运行速率不高。
 - ③高级语言程序无法直接访问、控制硬件。

编译和解释

- 编译器（Compiler）和解释器（Interpreter）都是将某种高级编程语言写成的源代码转换成低级编程语言目标代码的电脑程序。
- 编译：源代码（source code）→预处理器（preprocessor）→编译器（compiler）→汇编程序（assembler）→目标代码（object code）→链接器（linker）→可执行程序（executables）。
- 解释器是一个执行程序的虚拟机，一次能够翻译高级语言程序的一段、一行、一条命令或一个子程序。
- 解释技术更容易进行开发和调试，但开销很大；编译技术则可以产生更高效的代码，能够更加有效使用内存，对代码进行优化，程序执行更快。

软件开发

- 1950's年代，计算机主要用于科学计算，程序员主要是硬件工程师和部分科学家。
- 1960's年代以后，计算机逐步应用到企业和各种商业环境中，数据处理和事务计算成为了重要的工作内容，而这时的程序员还没有多少经验和原则可以遵循，这时的软件开发是“工艺式”的。“个人英雄主义编程”。
- 1960's年代后期，“软件危机”。

“软件工程”

- “软件工程”一词1968年首先在NATO（北约）组织的一次会议上作为正式的术语出现，标志着这一新的学科的开始。
- 会议组织者Brian Randell：“我们特意选择‘软件工程’这个颇具争议性的词，是为了暗示这样一种意见：软件的生产有必要建立在某些理论基础和实践指导之上——在工程学的某些成效卓著的分支中，这些理论基础和实践指导早已成为了一种传统。”

“工艺式”的软件开发的问题？

- 这是因为“工艺式”的软件开发难以解决软件开发的高度复杂性和对软件质量的要求。
- “狗窝”：“摩天大楼”
- 摩天大楼的复杂度和质量要求远比狗窝要高的多。如果要应对更复杂、更高质量要求的软件开发，我们也必须采用工程化的方法来开发，这样才有可能取得成功。
-

定义

- 软件工程作为一个新兴学科，一直以来都缺乏一个得到大家公认的定义。
- IEEE在软件工程术语汇编中的定义
- 软件工程是：
 - ①将系统、规范、可度量的方法应用于软件的开发、运行和维护，即将工程应用于软件。
 - ②对①中所述方法的研究。

- 计算机科学技术百科全书的定义如下：
- 软件工程是应用计算机科学、数学及管理科学等原理，开发软件的工程。软件工程借鉴传统工程的原则、方法，以提高质量、降低成本。其中，计算机科学、数学用于构建模型与算法，工程科学用于制定规范、设计范型（Paradigm）、评估成本及确定权衡，管理科学用于计划、资源、质量、成本等管理。

SWEBOK 2004

- SWEBOK指出软件工程包括十大知识领域：
 - 软件需求
 - 软件设计
 - 软件构造
 - 软件测试
 - 软件维护
 - 软件配置管理
 - 软件工程管理
 - 软件工程过程
 - 软件工程工具与方法
 - 软件质量

“软件工程道德规范和专业实践”

- 第一条原则中就提出软件工程师的一切行为原则的总纲：不能违背公众利益。
- “计算机及其相关技术正逐渐成为推动政府、教育、工业、商业、医疗、娱乐和整个社会发展的核心技术，软件工程师正是通过亲身参加或者教授软件系统的分析、说明、设计、开发、授证、维护和测试等实践工作，为社会做出了巨大贡献。也因为他们在开发软件系统中所起的重要作用，软件工程师有很大机会去为社会做好事或者给社会带来危害，有能力让他人以及影响他人为社会做好事或者给社会带来危害。为了尽可能确保他们的努力应用于好的方面，软件工程师必须做出自己的承诺，使软件工程师成为有益的和受人尊敬的职业，为了符合这一承诺，软件工程师应当遵循下列职业道德规范和实践要求。”