软件工程基础 —— 第2章 软件工程

计算机学院 孟宇龙

目录

2.1定义软件工程科学
2.2软件过程
2.3软件工程实践
2.4软件开发神话
2.5这一切是如何开始的

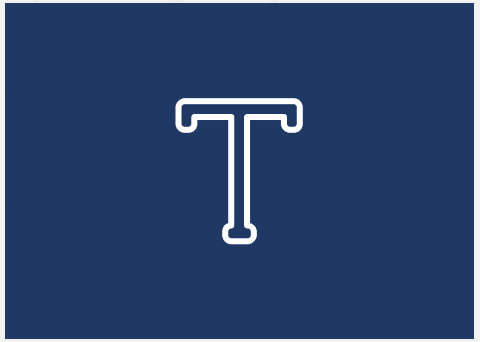
关键概念

- 框架活动
- 通用原则
- 原则
- 问题解决
- SafeHome
- 软件工程：定义、层次、实践
- 软件神话
- 软件过程
- 普适性活动

Point

- 在制定解决方案之前要理解问题
- 设计是一项关键的软件工程活动
- 质量和可维护性都来自于良好的设计
- 软件工程包括过程、管理和构建软件的方法和工具

软件工程



若干事实:

- 在制定解决方案之前要理解问题
- 设计是一项关键的软件工程活动
- 软件必须保证高质量
- 软件需具备可维护性

种子定义（Fritz Bauer）:

- （软件工程是）建立和使用一套合理的工程原则，以便经济地获得可靠的、可以在实际机器上高效运行的软件。
  未提及软件质量，直接谈到用户满意度或按时交付产品的要求、忽略了测量和度量的重要性和有效的软件过程的重要性。

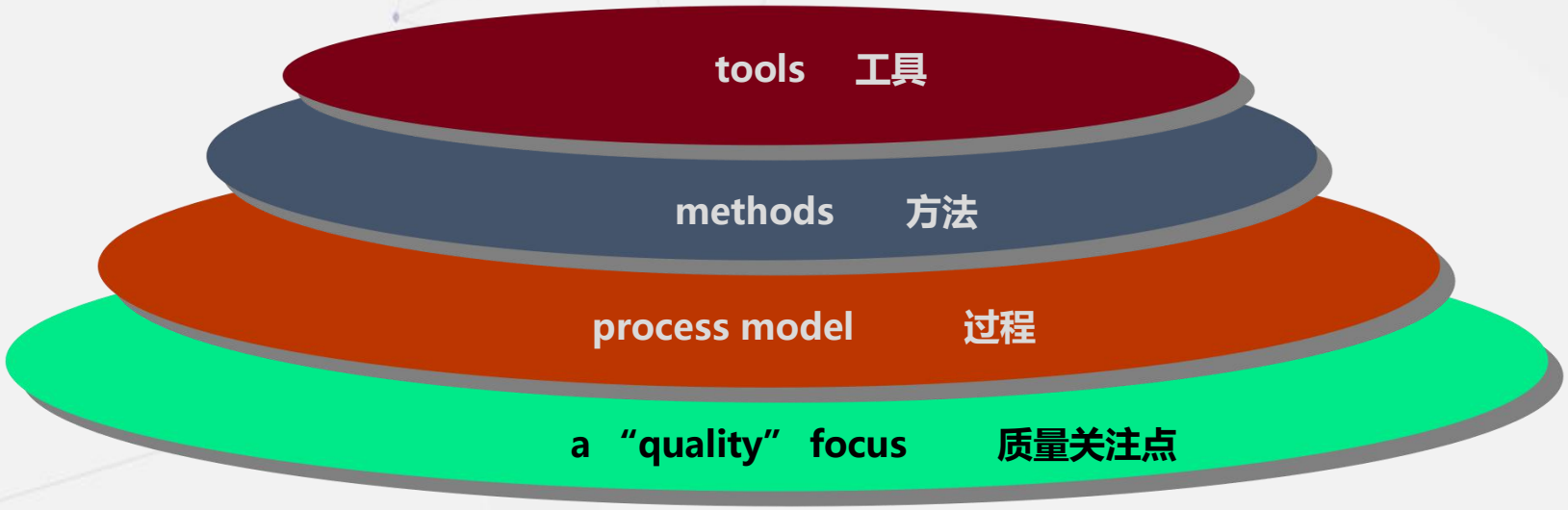
各种形式、各个应用领域的软件都需要软件工程

2.1 软件工程的定义

- IEEE 定义 软件工程是：
  （1）将系统化的、规范的、可量化的方法应用于软件的开发、运行和维护，即将工程化方法应用于软件。
  （2）在(1)中所述方法的研究。

-需要规范，也需要可适应性和灵活性

软件工程—种层次化技术



Software Engineering

软件工程层次图

一种层次化技术（续）

任何工程方法必须构建在质量承诺的基础之上。支持软件工程的根基在于质量关注点。

过程是软件工程的基础。过程将各个技术层次结合在一起，使得合理及时地开发软件成为可能。过程定义了一个框架，构建该框架是有效实施软件工程技术必不可少的。

方法为构建软件提供技术上的解决方法。包括沟通、需求分析、设计建模、编程、测试和技术支持。

工具为过程和方法提供自动化或半自动化的支持。 如CASE。

2.2 软件过程

软件过程是工作产品构建时所执行的一些列活动、动作和任务的集合

2.2.1 过程框架

过程框架（process framework)

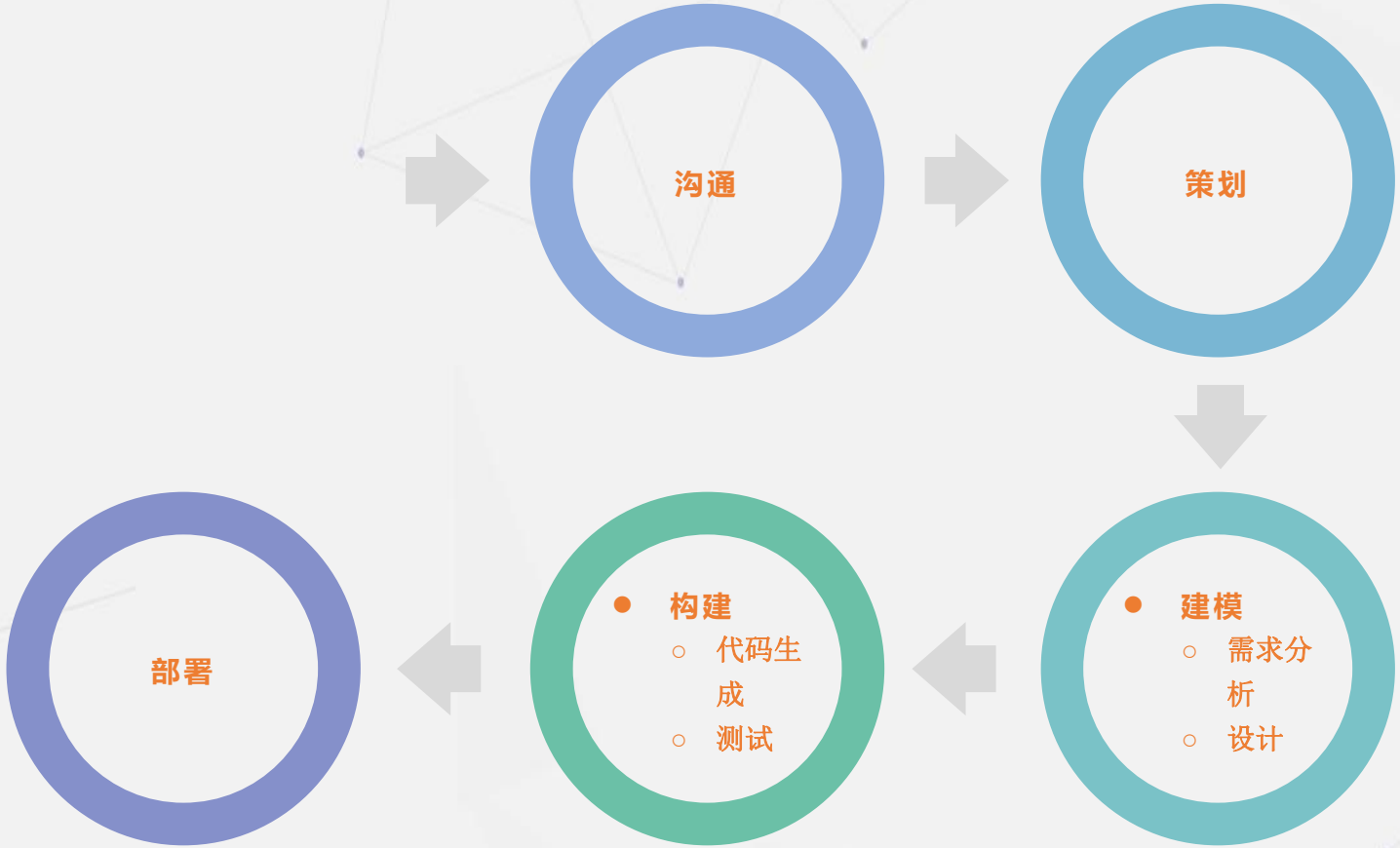
框架活动

- 工作任务(task)
- 工作产品
- 里程碑和可交付成果
- QA 检查点

普适性活动(umbrella activity)

一种过程框架

五个基本的框架活动



2.2.2 普适性活动

- 软件项目跟踪和控制
- 风险管理
- 软件质量保证
- 技术评审
- 测量
- 软件配置管理
- 可复用管理
- 工作产品的准备和生产

过程模型之间的不同

- 活动、动作和任务的总体流程，以及它们之间相互依赖关系
- 在每一个框架活动中，动作和任务细化的程度
- 工作产品的定义和要求的程度
- 质量保证活动应用的方式
- 项目跟踪和控制活动应用的方式
- 过程描述的详细程度和严谨程度
- 客户和利益相关者对项目参与的程度
- 软件团队所赋予的自主权
- 队伍组织和角色明确程度

2.3 软件工程实践

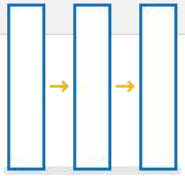
2.3.1 实践的精髓

1.理解问题

（沟通和分析）

2.计划解决方案

（建模和软件设计）



3.实施计划

（代码生成）

4.检查结果的准确性

（测试和质量保证）

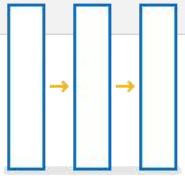
理解问题

谁将从问题的解决中获益？

也就是说，谁是利益相关者？

什么是未知的？

哪些数据、功能、特征和行为是解决问题必需的？



问题可以划分吗？

是否可以描述为更小、更容易理解的问题？

问题可以图形化描述吗？

可以建立分析模型吗？

计划解决方案

以前曾经见过类似问题吗？

在潜在的解决方案中，是否可以识别一些模式？是否已经存在有软件实现了所需要的数据、功能、特征和行为？

类似问题是否解决过？

如果是，解决方案所包含元素是否可以复用？

可以定义子问题吗?

如果可以，子问题是否已有解决方案？

能用一种可以很快实现的方式来表述解决方案吗？

能构建出设计模型吗？

实施计划

- 解决方案和计划一致吗？源码是否可追溯到设计模型？
- 解决方案的每个组成部分是否可以证明正确？设计和代码是否经过评审？或者更好的算法是否经过正确性证明？

检查结果

- 能否测试解决方案的每个部分？是否实现了合理的测试策略？
- 解决方案是否产生了与所要求的数据、功能、特征和行为一致的结果？是否按照项目利益相关者的需求进行了确认？

2.3.2 通用原则

1: 存在价值
2: 保持简洁
3: 保持愿景
4: 关注使用者
5: 面向未来
6: 计划复用
7: 认真思考

2.4 软件开发神话

管理神话

我们已经有了一本写满软件开发标准和规程的宝典。难道不能提供我们所需要了解的所有信息吗？ （是否已采用？开发人员知道不？是否反映了现状？是否全面？是否适应不同应用环境？是否在缩短交付时间的同时关注产品质量？）

如果我们未能按时完成任务，可以通过增加程序员人数而赶上进度。（蒙古游牧概念）（新人需要老人牺牲项目时间来培训）

如果决定将软件外包给第三方公司，就可以放手不管，完全交给第三方公司开发。（若开发团队部了解如何在内部管理和控制软件项目，就会在外包项目中遇到困难）

有了对项目目标的大概了解，便足以开始编写程序，可以在之后的项目开发过程中逐步充实细节。（需求描述不清楚会给项目实施带来灾难）

虽然软件需求不断变更，但是因为软件是弹性的，因此可以很容易地适应变更。（变更提出的时机不同，影响也不同，越早代价越小）

若干从业者神话

- 当我们完成程序并将其交付使用之后，我们的任务就完成了。
（60%-80%的工作耗费在软件首次交付用户之后）
- 直到程序开始运行，才能评估其质量。
（技术评审比测试有效，发现错误还早）
- 对于一个成功的软件项目，可执行程序是唯一 可交付的工程成果。
（各种工作产品（模型、文档、计划）是软件工程实施的基础）
- 软件工程将导致我们产生大量无用文档，并因此降低工作效率。（软工的目的是为了保证产品质量，减少返工，加快开发进度）

软件神话

- 影响管理者，客户(和其他非技术性的利益相关者）和从业人员
- 被认为是可信的，因为它们有时包含真实的部分

但是 …

- 不可避免的导致错误的决策

因此 …

- 按照正确理解软件工程的方式从实际出发解决问题

2.5 这一切如何开始

每个软件项目都来自业务需求——

- 对现有应用程序的纠错；
- 改变遗留系统以适应变化的业务环境；
- 扩展现有应用程序功能和特性；
- 开发一种新的产品、服务或系统。