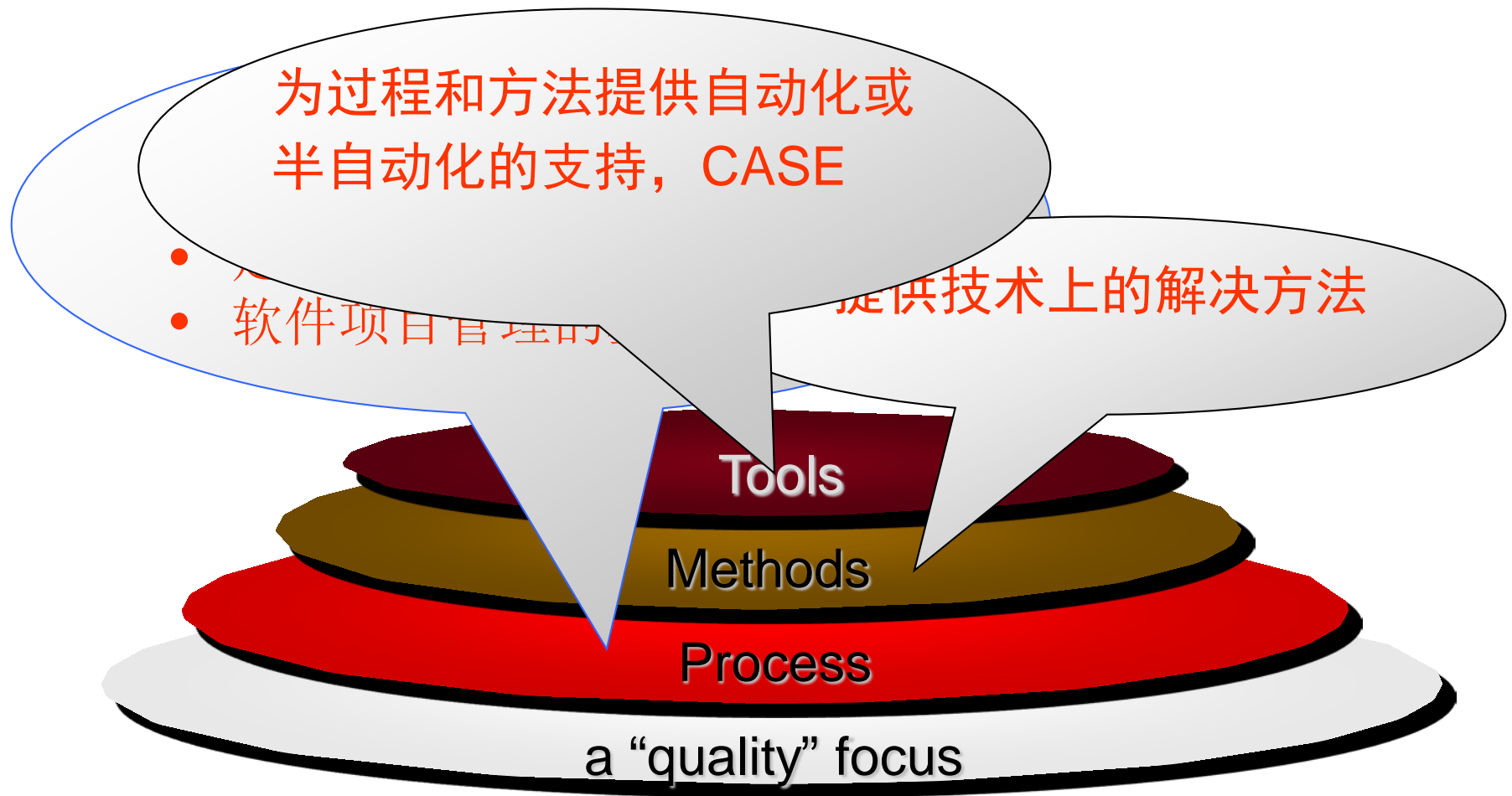


A Layered Technology



软件工程实践的精髓

■ Polya给出的建议：

□ 理解问题 (沟通和分析)

- 谁将从问题的解决中获益？即谁是利益相关者？
- 有哪些是未知的？哪些数据、功能和特征是解决问题必需的？
- 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？

软件工程实践的精髓

□ 策划解决方案 (建模和软件设计)

- 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？是否已经有软件实现了所需要的数据、功能和特征？
- 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描述解决方案吗？能构建出设计模型吗？

软件工程实践的精髓

□ 实施计划(代码生成)

- 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？或者采用更好的方式，算法是否经过正确性证明？

□ 检查结果的正确性 (测试和质量保证)

- 能够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需求的数据、功能和特征一致的结果？是否按照项目利益相关者的需求进行了确认？

Hooker软件工程实践的原则

- 1: 存在价值
 - 为用户提供价值
- 2: 保持简洁
 - 设计尽可能简洁，不是过于简化
- 3: 保持愿景
 - 清晰的愿景是软件项目成功的基础
- 4: 关注使用者
 - 在需求说明、设计和实现中，牢记要让别人理解你所做的事情

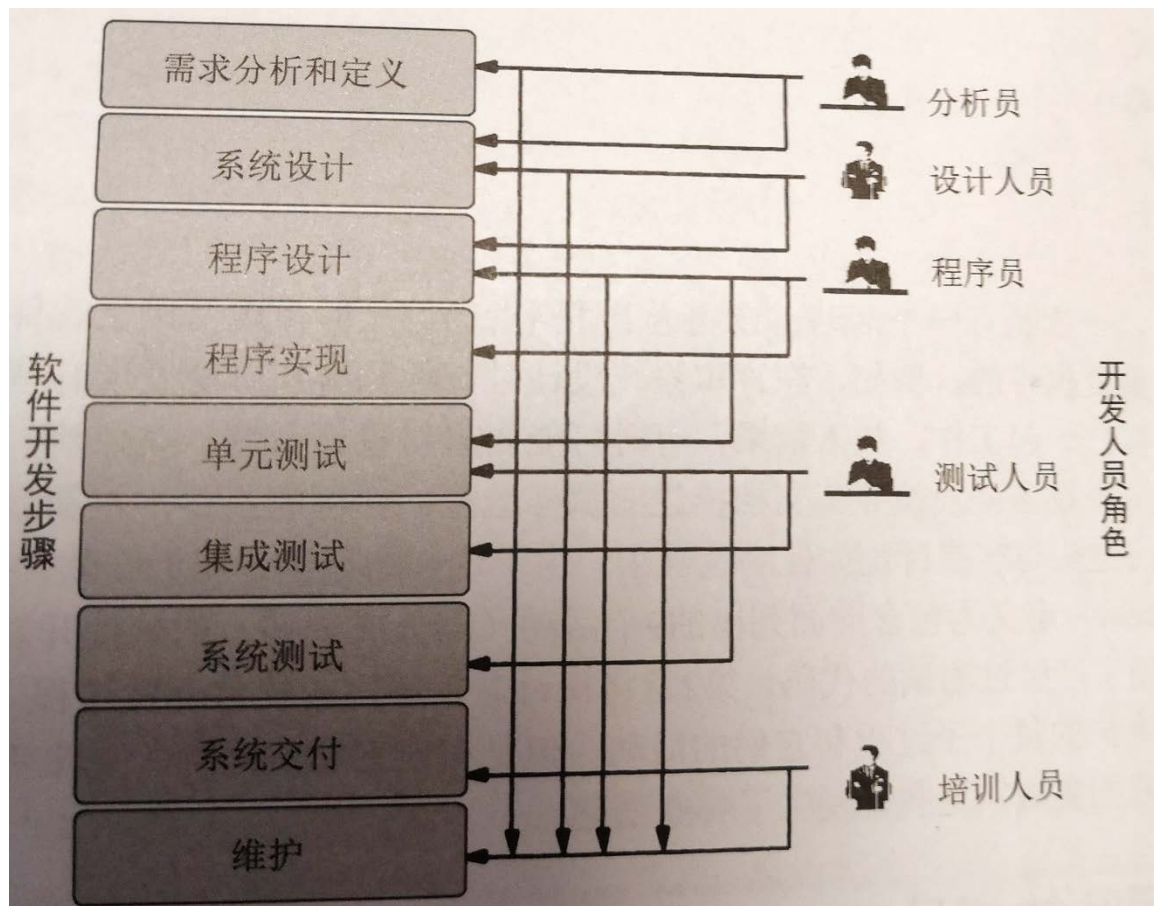
Hooker软件工程实践的原则

- 5: 面向未来
 - 系统能够适应各种变化，不要把自己的设计局限于一隅
- 6: 计划复用
 - 降低开发费用
- 7: 认真思考
 - 在行动之前清晰定位、完整思考通常能产生更好的结果

软件工程涉及的人员



软件工程涉及的人员



一切是如何开始的

- 每个软件工程项目都来自**业务需求**
 - 对现有应用程序缺陷的修正；
 - 改变遗留系统以适应新的业务环境；
 - 扩展现有应用程序功能和特性；
 - 开发某种新的产品、服务或系统。

软件工程道德规范

- IEEE与ACM联合制定
- 软件工程师应履行其实践承诺，使软件的需求分析、规格说明、设计、开发、测试和维护成为一项有益和受人尊敬的职业。
- 软件工程师不但需要熟练掌握软件工程的知识与技能，还必须要认识和遵循社会伦理和职业道德，具有强烈的职业责任感。
- 针对具体的事情，软件工程师应该根据自己的道义判断，最终做出合理的抉择。

软件工程道德规范

■ 八项原则

- ❑ **公众(public)**：软件工程师应当始终如一地以符合公众利益为目标；
- ❑ **客户和雇主(client and employer)**：在与公众利益保持一致的原则下，软件工程师应满足客户和雇主的最高利益；
- ❑ **产品(product)**：软件工程师应当确保他们的产品和相关的改进符合可能达到的最高专业标准；
- ❑ **判断(judgment)**：软件工程师在进行相关的专业判断时，应该坚持正直、诚实和独立的原则；

软件工程道德规范

■ 八项原则

- ❑ **管理(management)**：软件工程的管理和领导人员在软件开发和维护的过程中，应自觉遵守、应用并推动合乎道德规范的管理方法；
- ❑ **专业(profession)**：软件工程师应当自觉推动本行业所提倡的诚实、正直的道德规范，并自觉维护本行业的声誉，使软件行业更好的为公众利益所服务；
- ❑ **同行(colleagues)**：软件工程师对其同行应持平等互助和支持的态度；
- ❑ **自身(self)**：软件工程师应终生不断地学习和实践其专业知识，并在学习和实践的过程中不断提高自身的道德规范素养。

软件工程道德规范

■ 应遵守的规则

- ❑ 从不为了个人利益而窃取数据；
- ❑ 从不散布或售卖你所工作的软件项目的专利信息；
- ❑ 从不恶意地破坏或修改别人的程序、文件或数据；
- ❑ 从不侵犯别人、别的团队或组织的隐私；
- ❑ 从不为了某种利益而非法入侵别人的系统；
- ❑ 从不创建或传播计算机病毒。