

软件工程和架构的关系

起源

- 软件工程师一直在使用软件架构，但有时他们并未意识到这一点。
- 软件架构主要用来解决研究人员和从业者所识别出的关键软件工程问题，如软件重用等。
- 许多软件工程的思想源自其他领域，而不仅仅是计算领域。

软件工程困难

- 软件工程面临一系列独特的问题，需要适应非常复杂但无形的系统。
- 分为偶然困难和本质困难。

偶然困难

- 可以找到解决方案，可能是尚待发现的。
- 过去的生产力提高是通过克服编程构造和抽象的不足实现的。

本质困难

- 问题只存在部分解决方案，无法完全抽象化处理。
- 包括复杂性、一致性、可变性、无形性等问题。

复杂性

- 软件系统由众多互相关联的组件构成，使其难以理解、维护和扩展。
- 复杂性在规模增大时呈非线性增长，不同于物理产品。

一致性

- 确保软件系统与其他工程领域的变化相协调和一致是一个挑战。
- 软件必须适应外部需求的变化，需要持续的努力。

可变性

- 软件的需求和环境会发生变化，使得软件的持续维护和演化成为一项困难的任務。
- 变化源自新的应用程序需求、用户需求变化、新的硬件设备、标准变化和法律法规变化。

无形性

- 与物理产品不同，软件是无形的，增加了其开发、测试和维护的复杂性，因为问题不直观可见。

对抗复杂性的方法（1987 年）

- 购买 vs. 自建
 - 决定是构建还是购买，或购买哪部分，需要向客户展示产品以获取完整的规格说明。
- 增量/演进/螺旋式开发
 - 增长系统，而不是一开始就完全构建它们。
 - 有助于提高士气，容易回溯，提前制作原型。
- 杰出的设计者
 - 良好的设计可以教授，但伟大的设计不可复制。
 - 培养伟大的设计者。

设计的首要性

- 软件工程涉及需求收集、编码、测试、集成、配置等活动，但设计是贯穿整个工程活动的重要环节。
- 以面向架构的软件工程方法强调设计。

架构师的角色

- 架构师在项目中具有独特的角色和特征。
- 需要广泛的培训和丰富的经验。
- 具备敏锐的审美感和对领域的深刻理解。

万维网（WWW）简介

- 万维网是一组资源的集合，每个资源有一个独特的名称（URL）。
- 通信由客户端（用户代理）发起，使用 HTTP 协议。
- 万维网的架构是完全独立于代码的，存在多个代码片段用于实现各个组件。

架构实践：桌面

- 应用架构可以基于非常少的规则来理解，类似于 Unix 中的管道和过滤器。
- 应用程序可以由非程序员组成，类似于乐高积木。

产品线的实际案例

- 案例：内置 DVD 播放器的 35 英寸高清晰度电视机，适用于北美市场。
- 产品线的必要性和好处：重用结构、行为和组件，简化开发任务，减少时间和成本，提高系统可靠性。

总结

- 软件和建筑等其他工程制品都具有复杂性。
- 软件工程可以从其他领域中学到很多知识，同时需要拥有丰富的软件架构知识和经验。
- 架构的真正力量在于提供知识重用、智力控制、概念完整性和有效的沟通等优势。
- 重用带来了效率和可维护性的提升，同时降低了开发成本，增强了系统的可靠性。