



DeepL

订阅DeepL Pro以翻译大型文件。

欲了解更多信息，请访问www.DeepL.com/pro。

软件架构

鄂尔多斯市

北京交通大学

所有幻灯片的版权归北京交通大学软件工程学院鲍尔古德所有，任何传播都必须得到鲍尔古德的批准，否则就是违法。

联系我们

- 鲍尔古德，博士，教授
- 办公室。YF709
- 电子邮件：baoe@bjtu.edu.cn
- 电话。51684116
- 办公时间。通过预约

参考文献

- 《软件架构实践》第二版，作者：Len Bass, Paul Clements, Rick Kazman, 出版社。Addison Wesley, ISBN: 0-321-15495-9
- 《设计原则和设计模式》，作者：罗伯特 C.马丁，免费的在线图书，来自 www.objectmentor.com
- Java中的设计模式教程，来自 tutorialspoint.com 的免费在线书籍
- The Design Patterns Java Companion, By James W. Cooper, Free online book

- 这些幻灯片

评价

- 最终得分由三部分组成。
 - 书面考试50%
 - 课程项目40%
 - 见项目描述文件
 - 论文 10%
 - 关于本课程的知识结构的写作
 - 与课程项目一起上交

课程概述

- 软件架构简介 (2个单元)
- 软件架构的建模 (2个单元)
- 软件架构的质量属性 (2个单元)
- 设计原则 (8个单元)
- 建筑风格 (2个单元)
- 设计模式 (12个单元)

- 软件产品系列 (2个单位)

软件介绍

建筑学

鄂尔多斯市
北京交通大学

内容

- 软件开发的历史
- 软件架构的定义
- 软件架构的相关概念
- 对软件架构的影响因素
- 软件架构的好处
- 软件架构的研究与实践现状

软件开发的历史

软件开发的历史

- 组装语言
- 方案规模小

1970' ►

- 高级语言
- 以结构为导向的理论
- 数据流/控制流设计方法

1980'

►

- 应用程序开发库。类/函数库
- 面向对象的理论
- 对象建模与设计技术

1995 ►

- 应用开发框架。J2EE、.NET
- 组件技术。com/dcom, corba ...
- 对象建模和设计的标准化。UML

未来 ►

- 模型驱动开发。MDA
-

对软件历史的分析

- 软件的规模和复杂性越来越大，越来越复杂
 - 软件的应用领域：科学计算、制造、商业、教育和娱乐。
- 软件的抽象水平越来越高
 - 机器语言->组装语言->高级语言->应用框架
- 软件的设计正变得越来越复杂
 - 面向结构的编程 -> 面向对象的编程
编程->面向方面的编程

软件历史的结论

- 架构和设计比数据结构和程序算法更重要。
- 良好的架构设计一直是决定一个软件系统成功与否的主要因素

模拟



可由一个人建造 需要
最小的建模 简单的过
程 简单的工具

模拟



由一个团队最有效、最及时地建造
需要

建模

明确界定的流程 动力工具

模拟



模拟

- 不同之处在于
 - 规模
 - 过程
 - 费用
 - 时间表
 - 技能和发展团队
 - 材料和技术

- 利益相关者
- 风险

软件的定义 建筑学

现有的定义

- 一个程序或计算系统的软件架构是系统的结构，包括软件元素、这些元素的外部可见属性以及它们之间的关系。
 - "Software Architecture in Practice", Addison-Wesley, 1997年
- 架构是一个系统的组织结构。一个架构可以递归地分解为通过接口进行交互的部分、连接部分的关系以及组装部分的约束。通过接口交互的部分包括类、组件和子系统
 - UML 1.3
- 软件架构是一个系统的基本组织，体现在其组成部分、它们之间的关系和环境，以及指导其设计和演变的原则中。
 - ieee 1471-2000

对现有定义的分析

- 软件架构由以下部分组成
 - 软件元素：函数、接口、程序、类模块、层、子系统、客户/服务器等。
 - 属性：提供的服务、性能特征、故障处理、共享资源的使用等。
 - 关系：元素的构成机制和风格

简要定义


- 软件架构可以简要地定义为
 - 系统中包含的组件，以及这些组件的关系或互动机制
 - 组成部分：构成要素
 - 连接器。交互规则/机制
- 架构是一系列业务和技术决策的结果

软件架构设计

- 软件架构设计 = 分解
+ 构成
 - 降低软件设计和构建的复杂性
 - 控制软件开发的风险
 - 提高组织的效率和管理

系统





组成部分

软件架构设计

- 但是，我们必须考虑
 - 我们如何将这个系统分解成碎片？
 - 我们是否拥有所有必要的碎片？
 - 这些碎片是否合在一起？

更多定义

- CMU网页上有数百个定义 :
<http://www.sei.cmu.edu/architecture/definitions.html>

软件的相关概念 建筑学

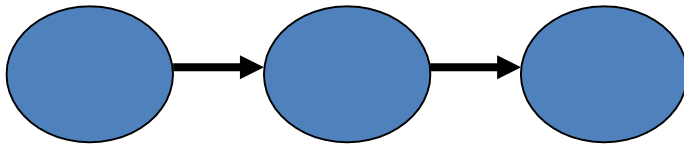
组成部分

- 服务于某些职责的系统的逻辑和功能单元
- 可能是不同情况下的不同具体对象
 - 模块、子系统、层、包、类等。
- 可分为较小的组成单位

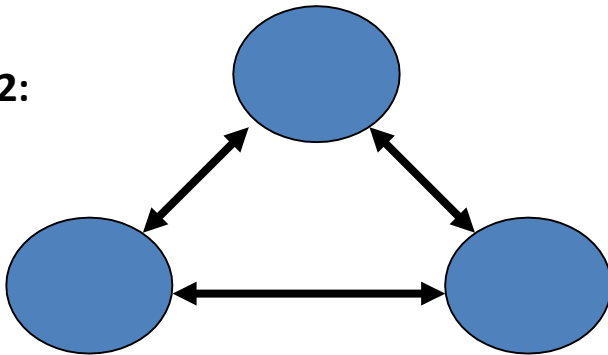
连接器

- 组件之间的互动规则或机制

SA1:



SA2:



SA3:



财产

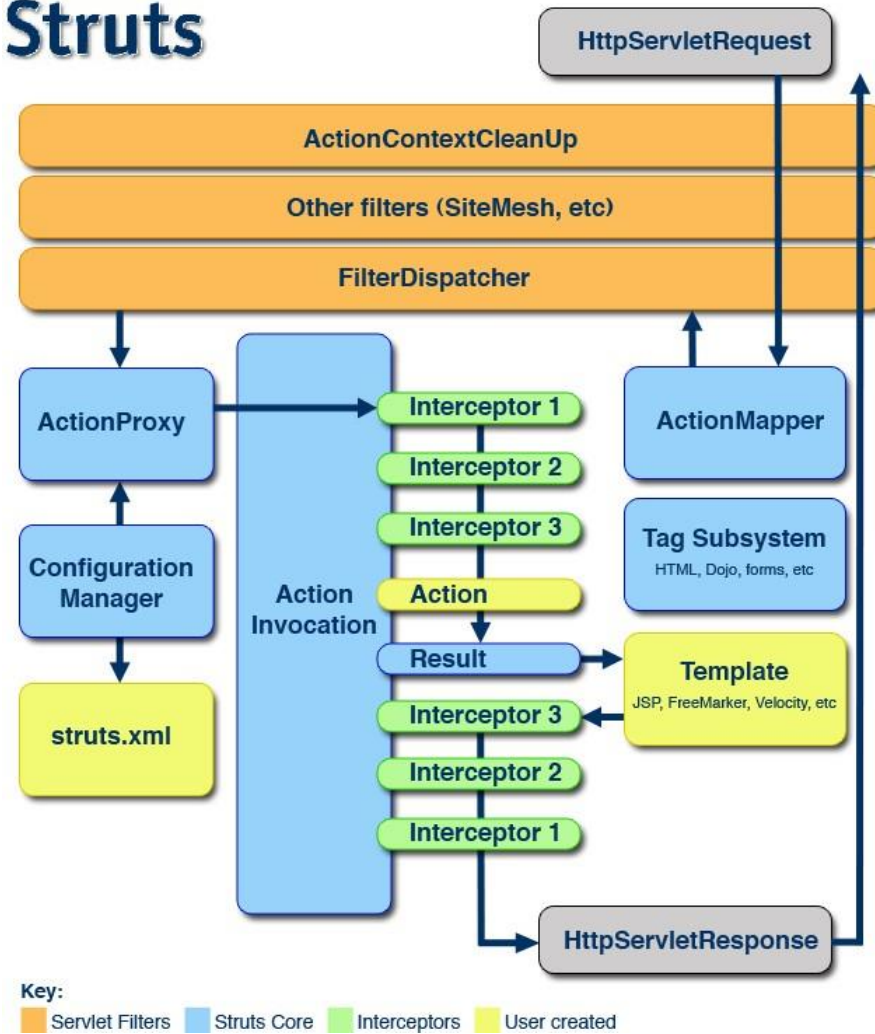
- 功能属性
 - 符合功能要求的特性
- 非功能性财产
 - 满足非功能要求的特征
 - 业绩
 - 便携性
 - 灵活性/可扩展性
 - 可靠性/安全性

框架

- 针对特定问题的可重复使用的应用基础设施
 - 有必要的的基本部件和连接器来解决指定的问题
- 在此基础上开发的应用程序的背景或环境
- J2EE框架、.Net框架等。

架构

Struts



对软件的影响因素 建筑学

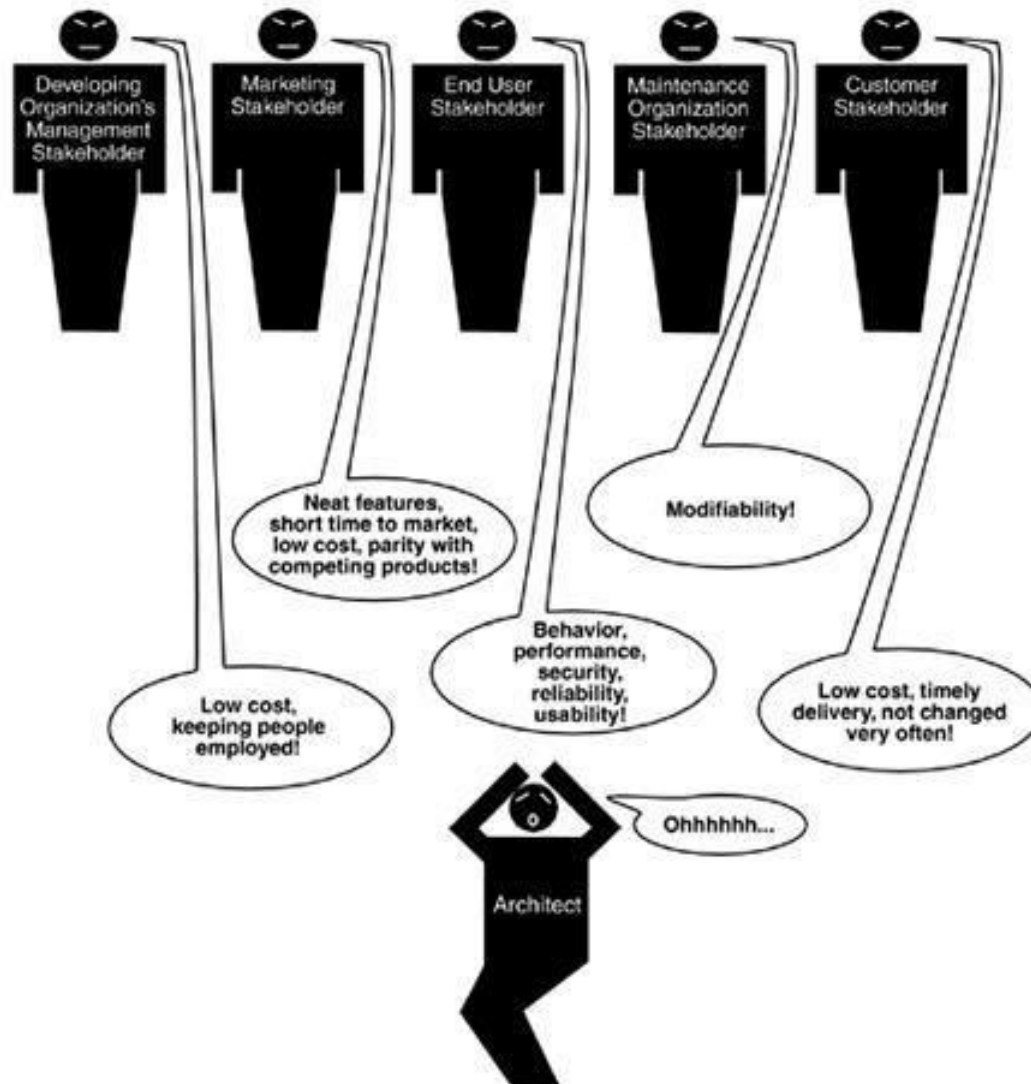
影响因素

- 系统利益相关者
 - 许多利益相关者，许多观点
- 发展中的组织
- 建筑师的背景和经验
- 技术环境

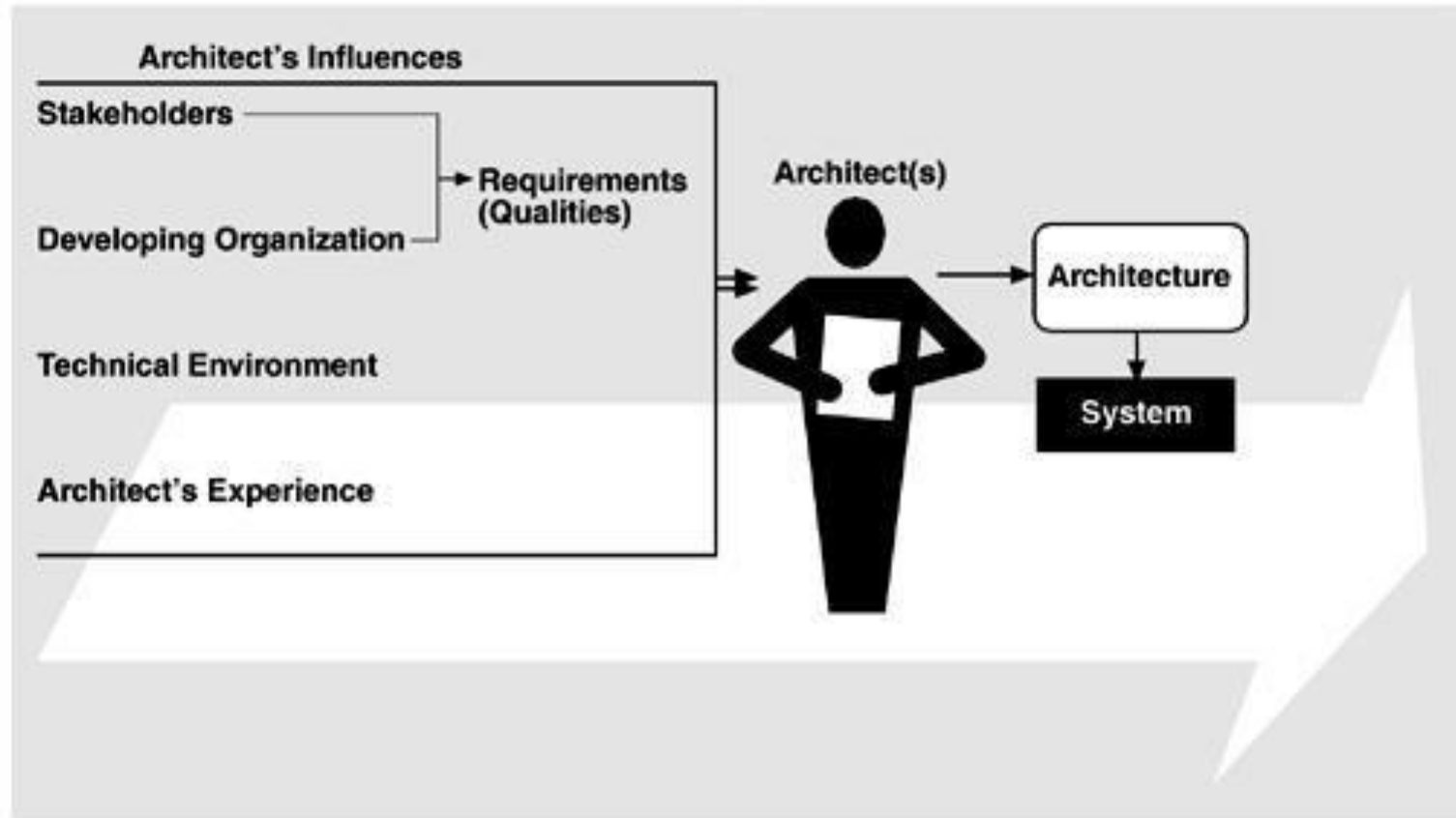
许多利益相关者， 许多观点

- 架构对于不同的利益相关者是不同的事情
 - 终端用户
 - 客户
 - 项目经理
 - 系统工程师
 - 开发商
 - 建筑师
 - 维护者
 - 其他开发商
- 多个利益相关者， 多个观点， 多个蓝图

Influence of stakeholders on the architect



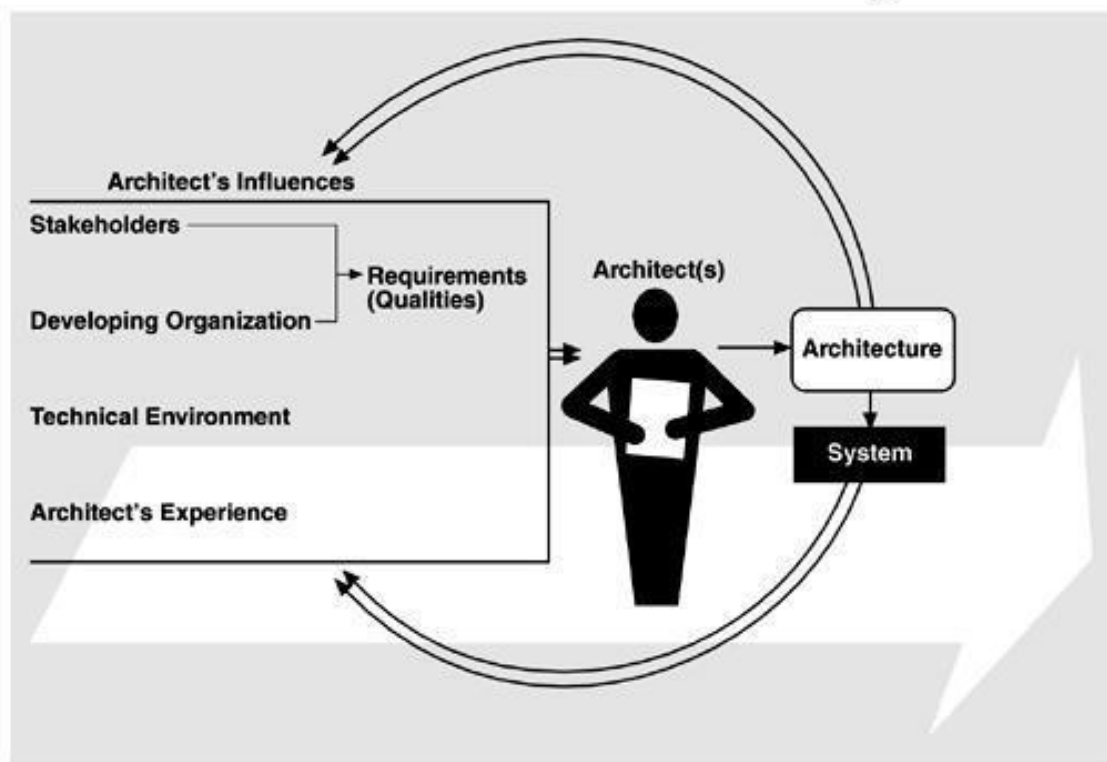
Influences on the architecture



反馈环路

- 建筑学对影响因素的影响是相反的

The Architecture Business Cycle



建筑学商业周期

- 为系统创建商业案例
- 了解需求
- 创建或选择架构
- 记录和交流架构
- 分析或评估架构
- 在架构的基础上实施系统

- 确保实施符合架构的要求

软件架构的好处

技术目的

- 满足系统要求和目标
- 指定实施和测试的约束条件
- 实现系统的灵活分布/分区
- 减少维护和进化的成本
- 增加重用，并与遗留和第三方软件整合

组织目的

- 为利益相关者提供高层信息
 - 成本和风险评估
 - 工作分配和项目进度
- 促进利益相关者之间的沟通

好建筑的特点

- 有弹性的
- 简单
- 平易近人
- 明确区分关注点
- 平衡的责任分配
- 平衡的经济和技术限制

目前的研究和实践在 软件架构

目前的研究和实践在 软件架构

- 形式化研究
 - 如何用具体的规则来描述软件架构
 - 架构描述语言（ADL）
- 验证和评估研究
 - 如何验证和评估一个软件架构是否满足所有的功能和非功能需求
 - 架构权衡分析法（ATAM）

目前的研究和实践在 软件架构

- 最新的研究来自于国际软件工程会议（ICSE） [: http://2015.icse-conferences.org](http://2015.icse-conferences.org)