软件架构产生的背景

软件开发效率低下,没有系统的开发理论体系,手工作坊生产,软件复杂度较高,软件产品质量差,技术手段滞后。

软件架构的作用

在高级层次上对软件系统进行描述,便于软件开发过程中各个视角的统一,能够 及早发现开发中的问题并支持各种解决方案的评估和预测。软件架构意义贯穿软 件生命周期的各个时期。

软件架构的思想

软件架构是软件系统的设计蓝图,不仅是软件系统的总体结构,也是一些质量属性以及功能与结构之间的映射关系,即设计决策。软件架构的两个焦点集中于系统的总体结构和需求实现之间的映射。软件架构的主要思想是讲注意力集中在系统结构的总体组织上。

软件架构特征

注重重用性;相关利益者多;关注点分离;质量驱动;概念完整性;循环风格。

软件架构发展阶段,各阶段特征

基础研究阶段--模块化;概念体系和核心技术形成阶段--组件化;

理论体系丰富发展阶段; 理论完善和普及应用阶段;

组成派与决策派定义

组成派关注软件本身,将软件架构看作组件和交互的集合,强调软件系统的整体结构和配置。

决策派认为软件架构是人们的设计决策,注重的是架构风格和模式的选择。

参考定义框架

组件;连接件;配置;端口;角色;

软件架构模型是什么

软件架构建模是对软件架构设计决策的具象化和文档化,它能够将软件架构的关注方面剥离出来,使用统一的图形、文档和数据进行描述,达到直观、便捷地理解、分析和交流。

软件架构五类方法

基于非规范地图形建模方法;基于UML的建模方法;基于形式化的建模方法;基于UML形式化的建模方法;其他建模方法;

软件架构建模方法的发展趋势

文本模型;图形可视化模型;UML模型;形式化模型;未来模型;

什么是软件架构风格

软件架构风格是在某一特定应用领域中系统组织方式的惯用模式。

使用架构风格的好处

可以极大促进设计的重用性和代码的重用性,使得系统的组织结构易被理解。采用标准的架构风格可较好的支持系统内部的互操作性以及特定风格的分析。

敏捷开发的基本理念

强调个人与交互比强调过程和工具好;

强调获得可运行软件比强调完善各种详尽文档好;

强调与客户合作比强调合同谈判好;

强调响应需求变化比强调遵循既定计划好;

敏捷开发与架构设计的关系

软件架构与敏捷开发的出发点是一致的,都是为了提高软件开发效率,提高软件 质量,降低成本;

敏捷开发也需要重视架构设计,架构设计对于敏捷开发也是必要的,两者在开发 实践中共同存在,相互促进。

敏捷开发改变了软件架构设计方式

敏捷开发将软件架构设计分为种子结构设计和详细结构设计;

种子结构设计描述了系统的总体框架;

详细结构设计在敏捷开发中被分散到整个敏捷开发软件过程中,以达到提高效率,减少风险的目的,

两类常见的敏捷软件架构设计方法

规划式设计和演进式设计,具体表现在初始阶段设计与迭代阶段设计。

架构驱动的软件开发步骤

架构需求获取;基本架构设计;架构记录文档化;架构评估;架构实现;架构维护;

架构的结构

基于功能;基于代码;并发;物理;基于开发;

成功的软件架构应具备的品质

良好的模块化;适应功能需求的变化;对系统的动态运行有良好的规划;对数据有良好的规划;明确、灵活的部署计划;

将软件架构概率引入软件需求阶段的优劣性

缺点:缺乏软件架构思想的软件需求,将会导致在架构阶段,软件需求规约与架构的映射相对困难:

优点:有助于保证需求规约,系统设计之间的可追踪性和一致性,有效保持软件质量,同时也可以获得更有结构性和可重用性的需求规约。

软件架构和软件需求如何协同演化

软件需求和软件架构两者是相辅相成的关系,软件需求影响软件架构,软件架构帮助需求分析的明确和细化,是一个相互影响的螺旋过程。

软件架构映射到详细设计的问题及解决办法

问题:缺乏重要构架视图,片面强调功能需求;不够深入,没有明确的技术蓝图;名不其实的分层架构,缺少交互接口交互机制,知识进行了职责划分;某个部分过度设计;

解决办法:补全缺少的架构视图;需要将设计决策细化到和技术相关的层面;步步深入,明确各层之间的交互接口和交互机制;切忌过度设计;

MDA的基本思想及应用好处

基本思想:将软件系统分成模型和实现两个部分:模型是对系统的描述,实现是利用特定技术在特点平台或环境中对模型的解释。模型仅仅负责对系统的描述,与实现技术无关。这是模型的实现技术无关性。

好处:将模型与实现分离后,能够很好的适应技术易变性。由于实现往往高度依赖特定技术和特定平台,当技术发生迁移时,只需针对这种技术作相应的实现,编写相应的运行平台或交换工具。能够较好的应对实现技术发展带来的挑战。

架构设计原则

一般性原则(商业原则,数据原则,应用程序原则,技术原则等);关键性原则(关注点分离,单一职责原则,最少知识原则等);

软件架构评估的必要性

体系结构评估可以减少后期的测试和纠错开销;

评估时挖掘隐性需求并将其补充到设计的最后机会;

体系结构是开发过程的中心,不良的体系结构会使软件开发一塌糊涂;

软件架构评估的方式分类

基于调查问卷或检查表的评估方式

基于场景的评估方式

基于度量的评估方式

4

4) 三种评估方式对比

评估方式	调查问卷或检查表		松里	度量
	调查问卷	检查表	捌尿	□ □
	通用	特定领域	特定系统	通用或特定领域
体系结构的了解程度	粗略了解	无限制	中等了解	精确了解
	早	中	中	中
	主观	主观	较主观	较客观
	体系结构的了解程度	请查问卷 通用 体系结构的了解程度 粗略了解 早	评估方式 调查问卷 检查表 通用 特定领域 体系结构的了解程度 粗略了解 无限制 早 中	调查问卷 检查表 物景 通用 特定领域 特定系统 体系结构的了解程度 粗略了解 无限制 中等了解 早 中 中

ATAM

敏感点, 权衡点;

- 1. ATAM方法陈述
- 2. 商业动机陈述
- 3. SA的陈述
- 4. 确定体系结构方法
- 5. 生产质量属性效用树
- 6. 分析体系结构方法
- 7. 集体讨论并确定场景优先级
- 8. 分析体系结构方法
- 9. 结果表述

SAAM

- 1. 场景的形成
- 2. 描述软件系统结构
- 3. 场景的分类和优先级划分
- 4. 间接场景的单独评估
- 5. 评估场景交互
- 6. 形成总体评估