[软件架构实践阅读笔记2](https://www.cnblogs.com/zlj843767688/p/13096938.html)

时间： 2020-06-01

本书的第二部分讲的是创建架构

设计师如何创建构架?这是本书第二部分要了解的重点。因为质量属性的实现对系统的成功至关重要，因此我们开始对质量属性以及设计师如何借助他的工具箱来实现质量属性进行讨论。

对质量好坏做出评判的通常是旁观者( Booth Tarkington的解释)。这对设计师来说意味着客户可能不喜欢某个设计方案，因为他们的质量属性概念与设计师不同。质量属性场景就是把质量属性从旁观者的角度转换为更加客观的基础的手段。在第4章，我们探讨了可能适合构架的不同类型的质量属性。对于6个重要的质量属性(可用性、可修改性、性能、安全性、可测试性和易用性)，我们描述了如何生成可以用来刻画质量需求的场景。

这些场景说明了对某个特定的系统来说质量意味着什么，它们为设计师和客户提供了一个对设计进行评判的基础。

　　当然，知道质量需求仅为设计师提供了一个目标。在第5章中，我们列出了设计师可以用来实现质量需求的战术和模式。例如，高可用性依赖于在数据或代码中某种形式的冗余，这种冗余需要设计师考虑额外的因素( 如确保复制的同步).

　　在第6章，我们介绍了第2个案例分析一一设计用来支持联邦航空局的空中交通管制功能的系统。该系统设计用来满足极高的可用性需求( 每年的停机时间要少于5分钟)，并说明了在第5章所列举的战术。质量属性场景和构架战术是一些可用于创建构架的工具。在第7章中，我们讨论了如何在设计构架和构建骨架系统的过程中应用这些工具，以及构架如何反映在组织的结构中。

　　在第8章，我们给出了关于飞行模拟器的第3个案例分析。这些系统被设计用来实现实时性能和可轻松修改性。本章说明了实现这些目标的方式。设计了构架后，必须将其编成文档。首先把相关视图编成文档，然后把所有特定视图都不能表达的内容编成文档。第9章详细阐述了构架编档。

　　我们通常得不到系统的构架:可能是因为根本没有把构架编成文档，或虽然编成了文档但丢失了，或现在的系统不同于所设计出来的系统。第10 章将介绍如何根据现有系统恢复构架。

质量属性

系统从设计、实现到部署的整个过程中考虑质量属性的实现。质量属性包括下列三类：

（1）、系统的质量属性。（可用性、可修改性、性能、安全性、可测试性和易用性）

（2）、受架构影响的商业属性。（上市时间、成本和收益、所希望的系统生命期的长短、目标市场、推出计划、与老系统的集成）

（3）、与架构本身相关的一些质量属性。（概念完整性、正确性与完整性、可构建性）

六个质量属性的战术列表：











