Chapter 15. Architectures in Agile Projects

敏捷开发中的架构

【了解：敏捷开发思想与准则】

**敏捷开发思想：**

应对变化

——变化比计划快、计划变化；

——不要墨守陈规、主动应对变化；

——测试驱动

**十二个敏捷原则：**

1. Highest priority is to satisfy the customer through early and continuous delivery.

最优先考虑的是通过早期和连续的交付来满足客户。

1. Welcome changing requirements, even late in development.

欢迎不断变化的需求，即使在开发后期。

1. Deliver working software frequently with a preference to the shorter timescale.

经常交付工作软件，优先于较短的时间范围。

1. Business people and developers work together daily throughout the project.

业务人员和开发人员在整个项目中每天一起工作。

1. Build projects around motivated individuals.

围绕有动力的个人建立项目。

1. Face-to-face conversation.

面对面的谈话。

1. Working software is the primary measure of progress.

工作软件是进度的主要度量。

1. Agile processes promote sustainable development.

敏捷过程促进可持续发展。

1. Continuous attention to technical excellence and good design.

持续关注卓越的技术和良好的设计。

1. Simplicity—the art of maximizing the amount of work not done—is essential.

简洁——最大化未完成的工作量的艺术——是必不可少的。

1. Self-organizing teams.

自组织团队。

1. At regular intervals, the team reflects on how to become more effective.

每隔一段时间，团队会思考如何变得更加有效。

【理解：敏捷开发的甜蜜点】

有两项活动可以增加项目进度:

1. 在体系结构和风险识别、计划和解决方面的前期设计工作
2. 由于修复缺陷和处理修改请求而返工。

直觉上，这两者相互抵消。

Boehm和Turner在三个假设的项目中绘制了这两个相对立的值:

一个10 KSLOC（KSLOC=千行代码）的项目

一个100 KSLOC的项目

一个1000 KSLOC的项目

这些行表示每个项目都有一个最佳点。

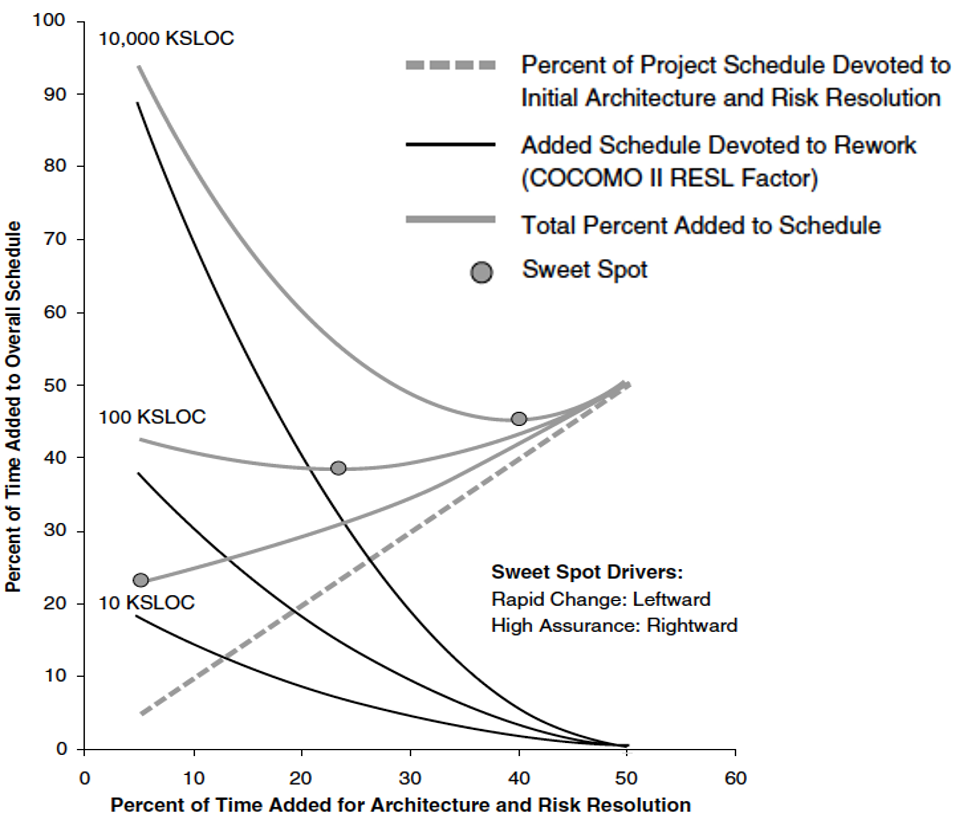
对于10KSLOC项目，最佳位置在最左边。对于一个小项目来说，投入大量的时间在前期工作上是一种浪费。

对于100 KSLOC项目，最佳时间点是项目进度的20%左右。

对于1000 KSLOC项目来说，最佳时间是项目进度的40%左右。

一个有一百万行代码的项目是非常复杂的。

如果没有体系结构来指导和组织工作，很难想象仅凭敏捷原则如何处理这种复杂性。



Percent of Time Added to Overall Schedule

增加到总日程表的时间百分比

Percent of Time Added for Architecture and Risk Resolution

架构和风险解决所增加的时间百分比

Sweet Spot Drivers:

甜点司机：

Rapid Change: Leftward

快速变化：向左

High Assurance: Rightward

高保险：向右

Percent of Project Schedule Devoted to Initial Architecture and Risk Resolution

用于初始架构和风险解决的项目进度的百分比

Added Schedule Devoted to Rework(COCOMO II RESL Factor)

用于返工的附加进度表(COCOMO II RESL因子)

Total Percent Added to Schedule

加到计划中的总百分比

Sweet Spot

甜点（为总和曲线的极小值）

【了解：敏捷开发与架构编档。敏捷开发与架构演化】

架构文档：

文档要为读者而写！如果读者不需要就不要写。但请记住，读者可能是维护人员，也可能是还没参与项目中的新人。

架构演化：

* 架构评估工作是敏捷过程的一部分。满足涉众的重要关注点是敏捷哲学的基石。
* 我们的架构评估方法以架构权衡分析方法(ATAM)为例。
* ATAM确定了一组最重要的涉众关注点。
* 定制轻量级架构评估很容易。

【编者建议】

* 对于具有相对稳定和易于理解的需求的大型复杂系统，预先进行体系结构工作将会得到回报。
* 对于需求不稳定的大型项目，从快速设计候选体系结构开始，即使它遗漏了许多细节。
* 准备更改并详细说明此体系结构
* 对于需求不确定的小型项目，至少要设法就要使用的主要模式达成一致。不要在架构设计、文档或分析上花费太多时间。

【本章总结】

* 敏捷宣言和原则重视紧密结合的团队，不断地、频繁地交付工作软件。
* 敏捷过程最初用于时间框架短的中小型项目。它们很少用于大型项目，特别是分布式开发。
* 大型成功的项目需要敏捷和体系结构的结合。
* 敏捷架构师采取中间立场，提出一个初始的架构并使用它运行，直到它的技术债务变得太大，这时他们需要重构。
* Boehm和Turner发现项目有一个“最佳点”，在这个“最佳点”上，预先的架构规划是有回报的。