



## 像架构师一样去思考

作者：Gregor Hohpe | 译者：张卫滨 | 策划：丁晓昀

2024-09-24 · 北京 · 本文字数：5442 字 · 阅读完需：约 18 分钟

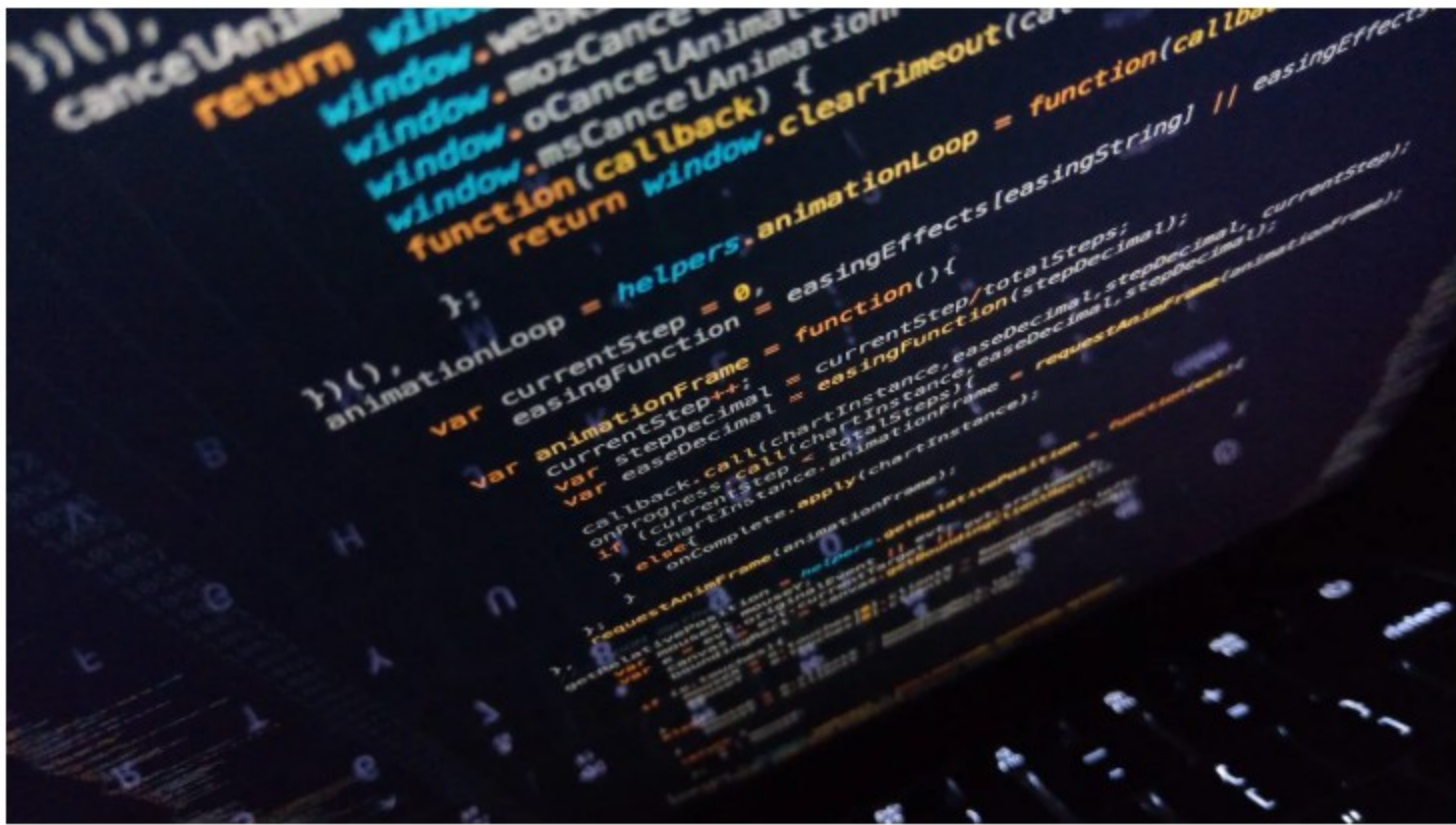


00:00

大小：2.66M

时长：15:29

1.0x



### 核心要点

- 架构师并不是团队中最聪明的人，他们是让其他人更聪明的人。架构师是智商的放大器。
- 乘坐架构师电梯意味着要连接楼顶和机房，现代架构师的价值是以他们能够覆盖多少个楼层来衡量的。
- 使用隐喻可以让观众更容易地进入思考。如果不邀请听众，那就是没有充分利用思维资源。
- 最强大的模型是最简单的。好的模型可以实现简化和抽象，提供清晰性，而不是造成困惑。
- 架构师要看到更多的维度：通过扩展问题和解决方案域，架构师能够让其他人更聪明地解决问题。

在 QCon London 2024 上，“《软件架构师电梯》(The Software Architect Elevator)”一书的作者 Gregor Hohpe 做了题为“[🔗像架构师一样思考](#)”的演讲。本文整理了他的演讲内容，他首先阐述了架构师的角色和连接层次（connecting level）的概念。

随后，演讲深入介绍了隐喻的重要性，它会使复杂的技术概念更加易于理解，并了解了利用模型做出更好的决策的优势。同时，我们还描述了从不同的角度处理问题、看到更多维度以及克服限制因素的重要性。

### 架构师的角色

架构师并不是由职称所定义的，它涉及各种视角和维度。有些架构师并不一定会在他们名片上印上“架构师”的字样，而那些拥有“架构师”头衔的人也不一定是优秀的。架构师更多的是一种思维方式和生活方式。

不同的组织对架构师的角色会有不同的看法。有些组织的运营可能没有正式指定的架构师，但是依然能够维持一致的架构。

需要明确，架构师的含义是复杂和多方面的。我们可以从不同的角度来思考架构师的作用，这对我们的工作很有助益。

### 架构会让其他人变得更聪明

关于架构师有一个最大的误解，那就是他们是最终的决策者，他们通常更为年长、收入更高，似乎也比其他人更聪明。但是，由一个人做出所有的重要决策是不现实的，因为没有一个人能够比所有人联合起来还要聪明。

架构师不能成为整个屋子里最聪明的人，并对别人发号施令。他们的角色是帮助其他人做出更好的决策，例如，从不同的角度看待问题、更好地理解权衡或者考虑更多的可选方案。这就是架构师作为团队的智商放大器所起到的作用：他们能够让其他人变得更聪明。

### 架构师连接各个层级

“《软件架构师电梯》”一书讨论了架构师在连接组织中不同层级的过程中的作用，并引出了哪个架构师对组织最有价值的问题。尽管有些人会认为首席架构师的价值最大，但是也有一些人认为是那些亲自动手实现解决方案的人。然而，在组织内部连接不同层级的架构师的重要性是不言而喻的。

首席架构师负责创建图表，如果这些图表脱离现实，那么他们的价值就会大打折扣。与之类似，经验丰富的开发人员的工作必须与组织更广泛的成就建立关联。这里的关键不在于开发人员所处的层次，而在于他们如何高效地驾驭和连接不同的层次。架构师电梯的概念强调了在不同的层级之间移动的重要性。

由于劳动分工、法律因素和结构化等级的原因，组织通常会划分为不同的层级，这往往会引发一些问题。位于“顶层”的领导层认为 IT 部门的一切工作都非常出色，并对区块链和 GenAI 技术夸夸其谈。与此同时，坐在工位上的开发人员却享受着自由的感觉，因为管理层并不了解他们的实际工作。中层管理人员为这种脱节的现象提供了便利，这个隔离层创建了一个松耦合的架构，但是在这种情况下，这并不

#### 推荐阅读

老架构师总结的 12 个软件架构陷阱 | 避坑指南

🔗 业务架构

区块链 + 社会治理：区块链如何为治理提供新方案？

2021-08-23

架构智能——下一代人工智能

🔗 AI&大模型

InfoQ 2024 年趋势报告：架构篇

🔗 架构, 云计算

智能合约：虚拟与现实的价值锚定载体

2021-08-09

困境与展望：区块链技术如何嵌入未来世界？

2021-08-25

只会坐在象牙塔？为数字化转型重新定义企业架构师的角色

🔗 数字化转型

#### 电子书



#### 中国开发者画像洞察研究报告 2024

分析开发者的行为模式、工作价值、职业发展等内容，帮助整个行业生态更深入地理解开发者，为他们提供更精准...

立即下载

#### 大厂实战PPT下载

换一换



#### 解密业务中台的发展之路

张浩飞 | 字节跳动 业务中台 产品专家

立即下载

#### 云环境下的企业运维平台演进历程

朱超健 | 阿里巴巴 GTS/资深技术专家

立即下载

#### 利用 SLM 结合边缘设备构建 AIoT Agent

卢建晖 | 微软 高级云技术布道师

立即下载



是一件好事儿！



图 1：楼层过多的危险

这种脱节的危险性显而易见，因为项目与战略目标并不一致，战略家已经脱离了现实。最有价值的架构师就是那些能够弥合这个鸿沟的人。

如今，高速变化的环境需要我们识别出组织和技术之间的相似性。在与高层管理人员交流时，架构师应该避免过度简化，因为高层管理人员是知识渊博的领导者，尽管他们可能不了解像 Helm Charts 这样的技术细节，但是我们鼓励架构师提供有意义的细节信息，以帮助领导者做出更好的决策。架构师电梯理念的提出是为了统一组织中各个层级的理解，而不是自说自话。

尽管讨论高水平的自动化、云基础设施和 DevOps 这样的话题会让技术团队兴奋不已，但是 CIO 和 IT 主管更关心的是避免安全漏洞、确保高可用性和维持成本收益。Hohpe 概述了如何弥合这些差异：技术创新直接解决了 CIO 层级的优先事项，但是架构师要乘坐电梯将位于不同层级的点联系起来。例如，自动化可以确保一致的补丁级别，这进而又提高了安全性。

### 使用隐喻

架构师有一个连接这些点的强大工具，也就是隐喻。选择与行业相关的隐喻，比如，当在金融服务业工作时，使用金融相关的隐喻可以使复杂的技术概念更加贴近现实，从而吸引领导者参与决策。隐喻有助于消除隔阂，建立信任，而不要使用高深的技术术语去轰炸他们。这样就能让领导者参与深思熟虑的决策，而不是简单地去寻去他们的批准。使用隐喻可以让他们参与决策过程。

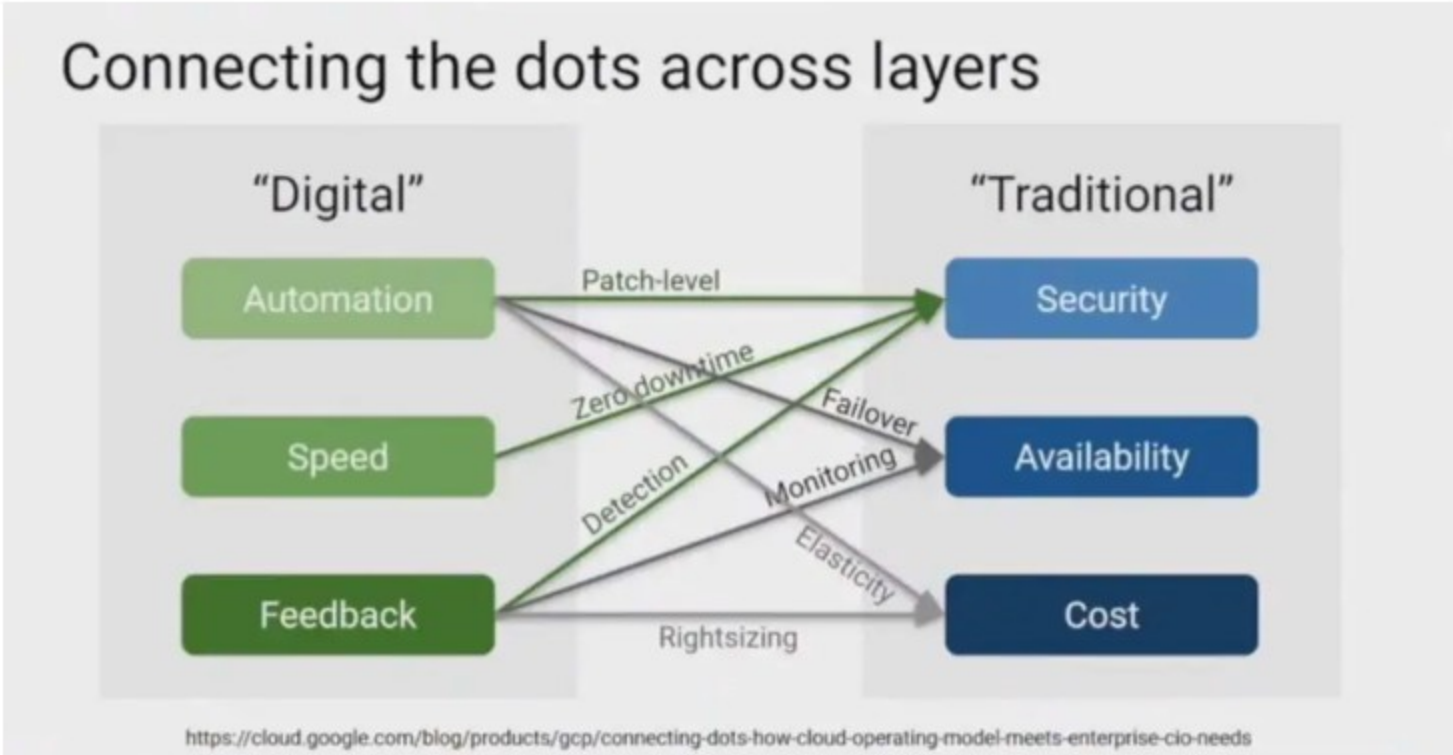


图 2：跨层连接不同的点

类似的，在与关注成本效益的管理者讨论交付速度时，提到频繁发布软件可能会让他们觉得不安。对他们来说，速度可以意味着忙中出错。为了消除这种隔阂，架构师应该阐明延迟成本的概念：部署延迟六个月可能会造成数百万的损失。这种方式将时间转换成了货币，克服了他们的心理障碍，更有助于理解。

### 架构师使用模型做出更好的决策

我们所处的世界并不简单。如今，我们所构建的应用之所以这么复杂，是因为它们是基于分布式系统、事件驱动架构、异步处理或横向扩展和自动扩展等能力的。虽然这些功能看上去非常高端，但是它们却增加了复杂性。

模型是架构师应对复杂性的最佳工具。模型的强大之处就在于它塑造了人们的思维方式。🔗[Dave Farley](#) 用一个例子阐明了这一点：很久以前，人们认为地球位于宇宙的中心，这种观点会让人觉得行星的运动是不稳定的，而且非常复杂。真正的问题不是行星的运动，而是采用了不正确的模型。当你把太阳放到太阳系的中心时，一切就变得合理了。  
当架构师向不同领域的人解释问题时，可能会认为别人不理解他，但实际上是因为他们使用了不同的心智模型。

🔗[George Box](#)有一句名言：“所有的模型都是错误的，但有些模型是有用的”。但是，我们经常忽略了这句话的后半部分：

正如设计出简单而令人赞叹的模型是伟大科学家的标志一样，过度复杂和过度参数化往往是平庸者的标志。

在架构中，图表和复杂的产出物（如数据库模式和系统架构）是很常见的。不过，这些模型并不一定总能提供太多的抽象性。最强大的模型就是最简单的模型。好的模型可以简化和抽象，提供清晰度，而不是带来混乱。



当有人说，“但是，这个模型并不完全符合现实”的时候，我们的回答应该是这样的：“模型本来就不是完全符合现实，它只是一个模型。它能够让我们的思维更清晰，让我们更聪明，帮助我们做出更好的决策。它能提高我们的智商”。当架构师试图捕获现实中的每个细节时，他们就陷入了 George Box 所说的过于复杂的陷阱了。

我们该如何简化模型呢？以地球为例，我们有四个不同的模型。哪一个是最好的呢？它们都很好，这取决于问题是什么。



图 3：最好的模型是由问题决定的

模型只有在能够帮助我们回答问题时才有其价值。需要徒步？那请使用地形图。想要参加选举？那请使用政治地图。想以最快的速度从 A 到达 B？请使用路线图。想了解物流密度？请使用人口地图。不同的问题需要使用不同的地图。当被问到，“给我看一下架构”的时候，你应该反问，“你的问题是什么？”——不同的问题需要不同的模型。

### 架构师能够看到更多的维度

架构师可以从多个维度看问题，从而让其他人更加聪明。通过扩展问题和解决方案空间，架构师可以让其他人更聪明地处理问题。通常，当双方从不同的角度看待问题时，就会产生分歧，这就像下图这样围绕正方形和三角形争论不休，毫无进展。

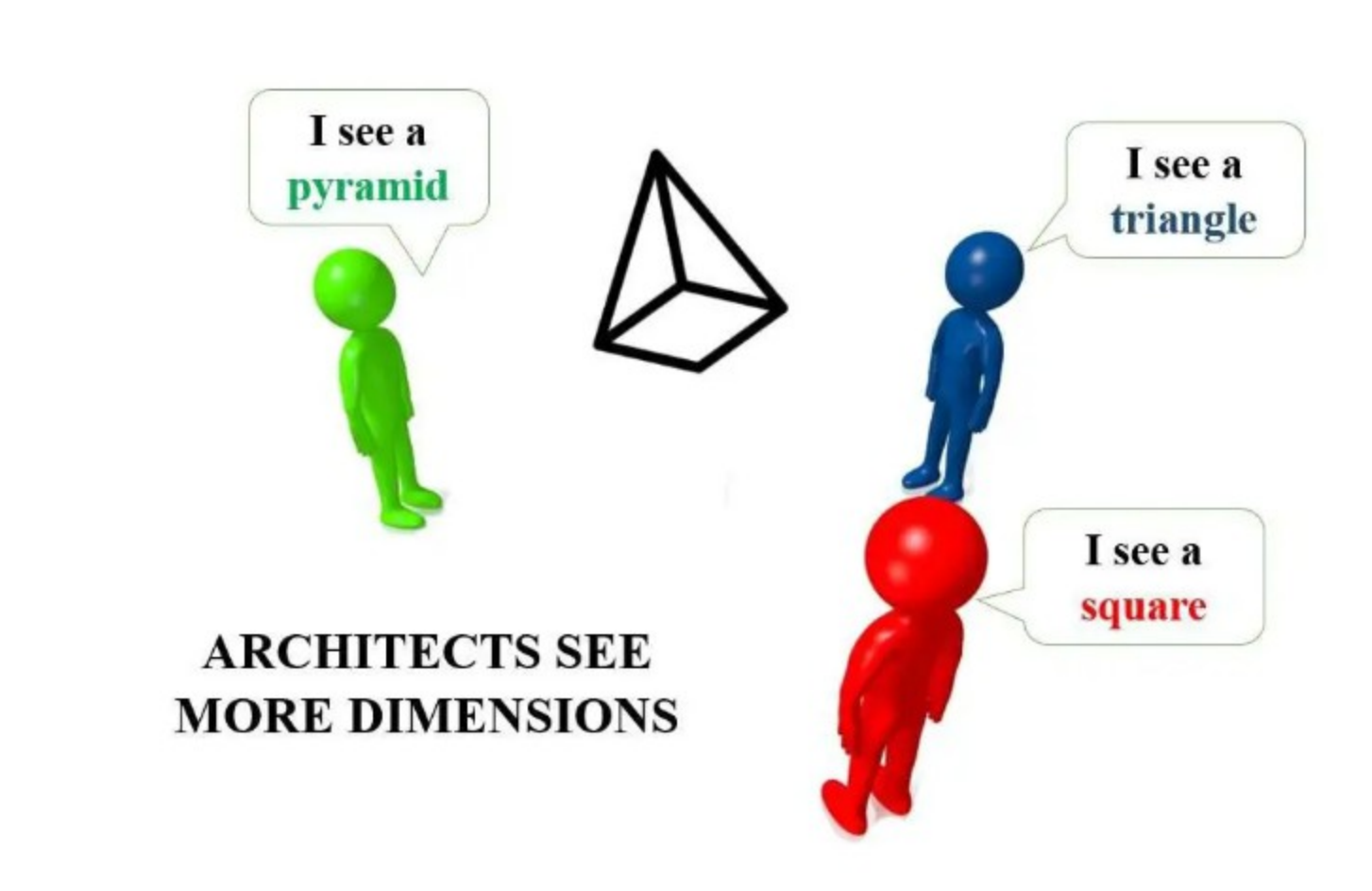


图 4：架构师能够看到更多的维度

架构师可以引入第三个维度，向大家展示对象其实是一个金字塔结构。例如，当团队在加快交付速度和保持质量之间争论不休时，架构师可以提出自动化测试或提前集成测试的方案，从而提升集体解决问题的能力，跳出非此即彼的辩论。

在 IT 战略领域，许多领导者都在面临“左右为难”的困境。一方面是标准化和降低成本以实现统一和规模经济的压力，但这会限制灵活性，因为开发人员可能更喜欢其他创新性的解决方案。另一方面，企业需要敏捷性和创新性来保持竞争力。这就形成了一种线性、不和谐的现象，双方都不满意。架构师如果能够识别出更多的维度，那么就可以更有效地应对这些挑战。

🔗平台可以实现协调和标准化，但不会扼杀创新，反而能够促进创新。这一概念已经不再局限于 IT 领域，而是延伸到了其他行业，比如，20 世纪 70 年代大众汽车采用平台模式时的汽车制造业。通过在一个通用平台上实现底层工程的标准化，他们能够通过多种模型实现车型的多样化和创新。这种方式不仅没有减少多样性，反而大大增加了汽车的品种。宝马从最初的三种车型发展到了如今的 30 多种车型。

摆脱单向思维至关重要：汽车公司通过平台标准化扩展了产品多样性，软件架构师也应如此。通过采用多维方式，可以在质量和速度之间取得平衡，并通过平台实现标准化和创新。对于寻求效率和创造力最大化的架构师来说，摆脱这种二元选择的思维方式非常重要。



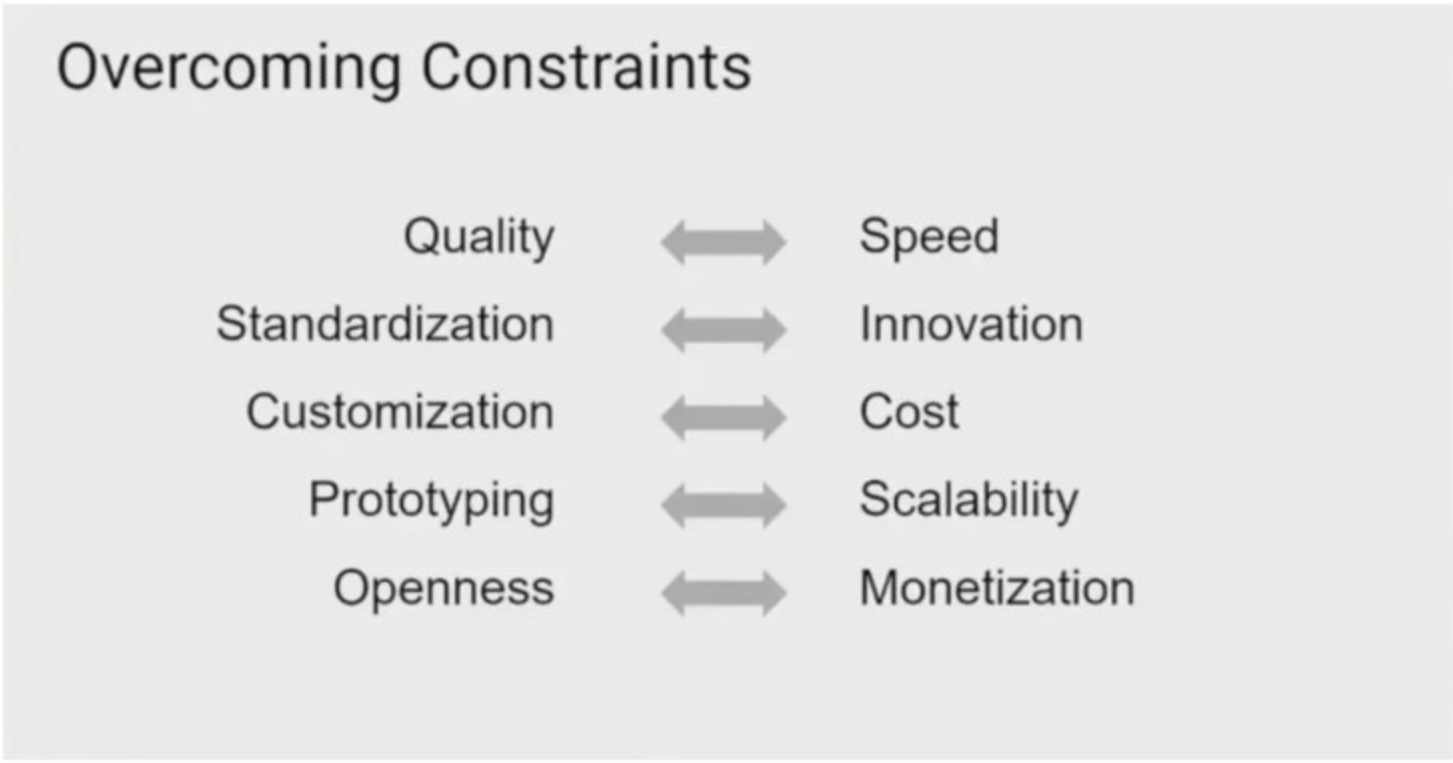


图 5：克服制约因素

另外一个增加思维度的样例就是供应商锁定和切换成本的问题，这是云供应商之间不太受欢迎的话题。通常情况下，对话会围绕在不同的云供应商之间切换工作负载所面临的挑战而展开，这在传统上会被视为一个单向思维的问题：决策通常会在两害相权之间做出选择，而无法找到一个中间点。更好的方式是在讨论中引入更多的维度。

例如，供应商锁定和切换成本可以视为潜在的负债，考虑到这种迁移的概率和规模，可以对成本进行量化。另一方面，无服务器架构和托管服务所带来的好处也能降低运营的复杂性。

通过考虑收益与切换的成本，可以更有效地权衡利弊，提供决策模型，而不是直接给出答案。这种方式可以让客户更好地了解其独特的限制因素和目标，使他们能够做出明智的决策。

客户说：我被锁定了。这听上去是一个非此即彼的状态：要么被锁定，要么没有被锁定。但实际上并非如此，这应该是一个可高可低的成本问题，它有不同的维度。

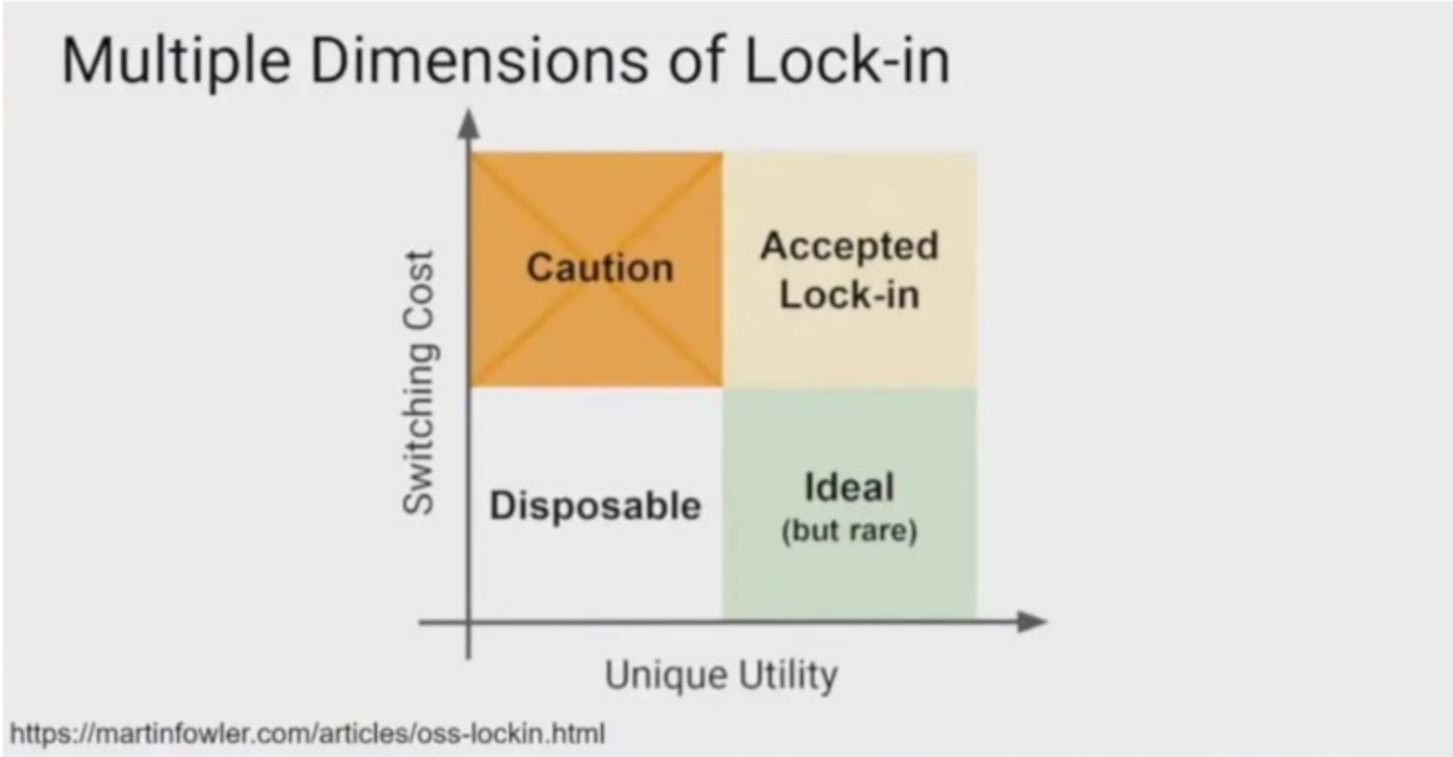


图 6：锁定的多个维度

锁定不仅局限于切换供应商。AWS 在其托管的 Kubernetes 服务上 EKS 上提供了扩展版本的支持。作为一名架构师，你希望每个人都使用最新版本，对吧？那为什么需要这样的服务呢？答案很简单：版本升级也是有成本的，由于切换成本的原因，你会被锁定到开源项目的某个特定版本上。架构师应该了解这些细微的差别，并考虑如何升级能够更具成本效益。

### 架构师兜售的是选择

很多人认为多样性和协调性是相互对立的。试想，开发人员对编程语言有不同的偏好。为了协调这些偏好，作为 IT 和软件架构师，我们使用像 OAuth 和 JSON 这样的协议创建标准 API 和接口。这种方式在微服务架构中很常见。

这里的关键在于，要将特定的元素标准化，如接口、授权协议、传输方法和数据表述格式。通过锁定这些元素，架构师牺牲了一些选择，却获得了另外一些选择，从而能够使开发人员使用各种语言进行编码。Hohpe 强调了协调性和灵活性之间的平衡，以便于同时实现创新和标准化。

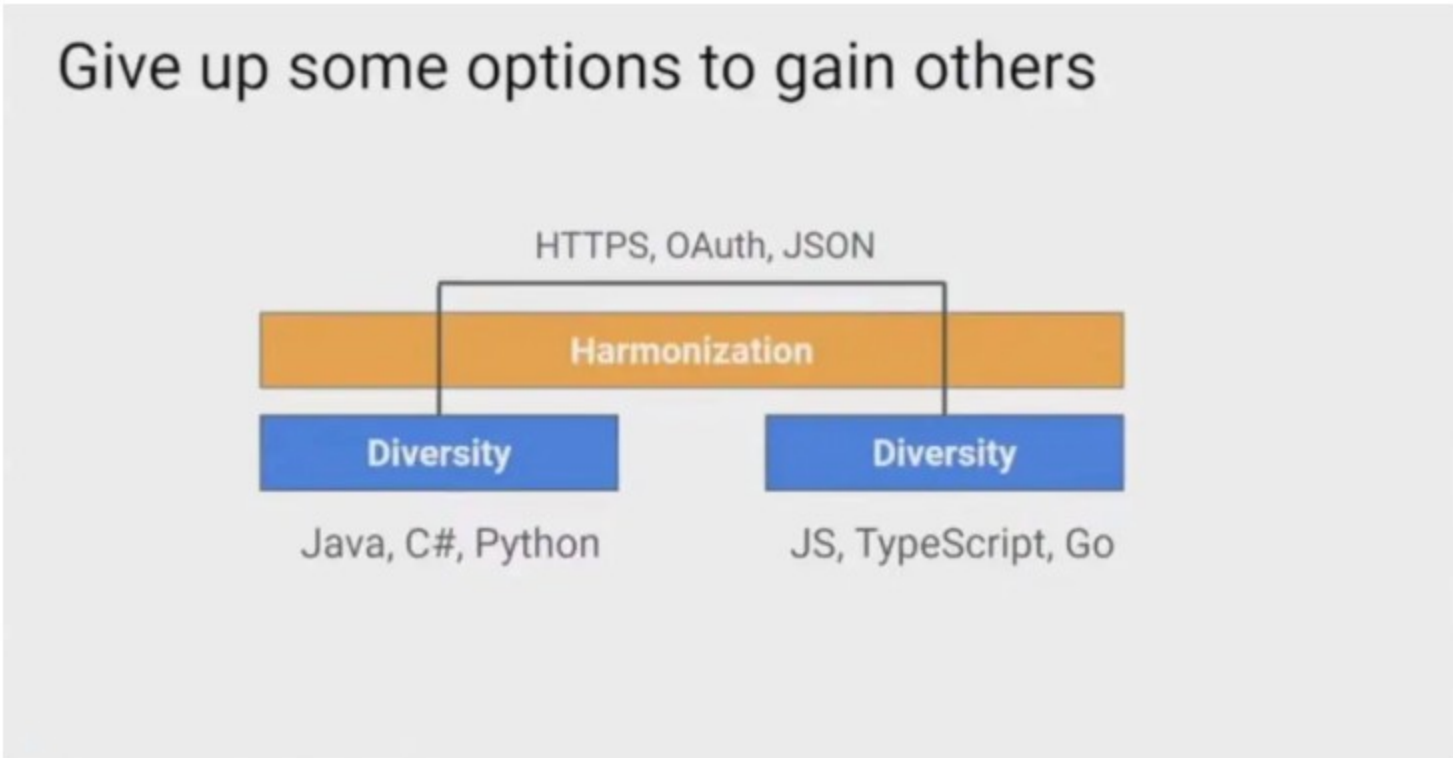


图 7：放弃某些选择，从而获取另外的一些选择

为了解释这种权衡，我们可以使用一些隐喻，尤其是来自金融领域的隐喻。将某些元素标准化以实现多样性类似于期权交易：软件从业人员放弃了一些选择，但是获得了另外一些选择。这些选择，比如使用不同的编程语言、将工作负载转移至另外的供应商、扩展系统和添加硬件容量，都是非常有价值的。

对于金融从业者来说，这个概念就像 Kubernetes 和 Helm Charts 对于 IT 专业人员一样清晰。期权定价（Black-Scholes 模型曾获得诺贝尔奖）表明，拥有期权总是有价值的。推迟决策，例如以后再利用云的弹性基础设施和扩展架构来增加容量，就是一种有价值的选择。

该公式的一个重要启示涉及到易变性：随着不可预测性的增加，期权会变得更更有价值。例如，如果用户数量稳定，就不需要弹性硬件。相反，如果在用户数量不确定的情况下推出移动应用或电子商务站点，那么扩展相关的选择就变得至关重要。不确定性越大，选择就越有价值。这个比喻说明，在架构领域，就像在金融领域一样，不确定性会增加选择的价值。使用这样的隐喻可以使复杂的想法更加清晰，更有说服力。

## 敏捷架构：是友非敌

架构和敏捷方法论都是在处理不确定性和易变性的基础上发展起来的。如果一切都是可预测的、一成不变的，那么就不需要敏捷性了。敏捷在变化中不断成长起来，就像架构一样：它们都能应对不确定性，是互相补充的。用汽车来类比的话，敏捷就是方向盘，用来指引方向，而架构则是发动机，确保汽车能够动起来。两者缺一不可，相互配合。在不断变化的世界中，要想取得成功，两者均不可或缺。架构和敏捷是盟友。

## 结论

在本文中，我们深入探讨了软件架构师的角色，反思了架构师是如何思考和决策的。我们讨论了架构师是谁以及要做什么的话题，重点阐述了使用模型和隐喻找到合适抽象的重要性。

我们还讨论了架构师作为智商放大器的概念，以及理解权衡和驾驭不同选择的重要性。

### 作者介绍

Gregor Hohpe 负责帮助技术领导者转型他们的组织和技术平台。你会发现他不断乘坐架构师电梯在机房到顶楼之间穿梭，也许上午他在设计自动化无服务器的解决方案，下午又在准备董事会的报告。Gregor 曾担任 AWS 和 Google Cloud 的 CTO 办公室负责人、新加坡政府的智能国家研究员以及 Allianz SE 的首席架构师。在 Allianz SE，他负责全球数据中心的整合架构，并部署了首个私有云的软件交付平台。Gregor 是开创性著作《企业级集成模式》（Enterprise Integration Patterns）的合著者，该书为所有现代 ESB 提供了参考。他的著作《软件架构师电梯》讲述了来自一线 IT 转型的故事。

### 原文链接：

[🔗Thinking Like an Architect](#)

发布于：2024-09-24 09:30 | 阅读数：12542  
文章版权归极客邦科技InfoQ所有，未经许可不得转载。

 架构

 轻点一下，留下你的鼓励

### 1 评论 1 条评论

写下你的想法，一起交流

发布



Fish

架构师是团队集体智商的放大器 👍

2024-09-24 10:27 · 广东

👍 0

💬 回复

没有更多了

### 1 更多内容推荐

#### 春节荐书（二） | 《人类简史》：故事的力量

今天我想跟你推荐另一本帮助理解复盘的书，《人类简史》，它的作者是尤瓦尔·赫拉利。

2021-02-15

#### 沟通中的冲突：什么时候应该妥协，什么时候应该坚持？

你如果用妥协来避免冲突，这绝对不能很好地解决问题。

2020-07-20

#### 资深架构师十年总结：成为架构师，你必须具备这五点能力

作者是一名从业数十年的软件架构师，在与不同领域、不同学科的软件工程师交流时，大家都会问，如何成为一名...

 架构, 方法论, 最佳实践, 性能优化, 团队搭建, 保险, 行业深度

#### 架构师实战营 模块二总结

架构师实战营 模块二 总结

2021-04-17

#### 架构思维：数字化转型如何落地？

数字化转型其实是企业从当前状态走向目标的过程，也需要在一定的指导原则下进行演进，所以，架构思维是必不...

2021-04-12

#### 架构师成功沟通的 3 个关键

架构师应该如何有效地沟通？本文提供了三个建议。

 架构, 文化 & 方法, 方法论, 汽车

#### 架构实战营 – 模块一作业

架构师实战营模块一作业

2021-10-26



9 年当上架构师，我的很多想法变了

“年轻时的我肯定会冲现在的我翻白眼。”

🔗文化 & 方法, 架构, 技术管理, 方法论, 微服务, 行业深度

论平台工程的价值

由于平台工程师的角色还比较新，所以对于这个角色的需求和所提供的价值还没有形成广泛的共识。

🔗架构, 服务革新, 软件工程, 研发效能, 技术选型, 操作系统, 框架, 微服务, DevOps & 平台工程

以流程驱动组织变革：实现多方共赢的局面

什么是好组织？本节课第一个重要思想就是告诉你流程决定组织。

2021-03-10

你的架构决策记录是否失去了它的目的？

架构决策涉及系统使用的基本概念，因为选择对代码的影响散布在整个软件中，而不是局部的。

🔗业务架构

架构的尽头是架构师

原来，架构的尽头是架构师，以一己洪荒之力对抗熵增，化无序为有序，还得保持情绪稳定，身心健康，动手时候...

2022-05-31

人力资源管理：以奋斗者为本

奋斗者如何定义？全面薪酬激励如何推行？员工的价值如何评估？

2021-03-10

避免成为“象牙塔”架构师：架构师和组织之间的关系

如何在理解细节和纵观全局之间取得平衡？

🔗架构, 方法论

敏捷时代的架构

在敏捷时代，架构如何适应敏捷原则，架构师如何以敏捷的方式与各个团队合作，本文希望能够给出这些问题的答...

2023-09-17

分段演进：数字化转型需要经历哪些阶段？

只谈当前的起点和未来的终点，可操作性不强，我们必须得清楚都有哪些阶段，拆解下咱们的转型过程。

2021-05-15

架构师实战营 – 模块四 – 作业

模块4作业

2021-11-21

架构师角色的演变：从发号施令到与团队合作

本文探讨了共享架构的文化和架构师角色的演变。

🔗架构, 文化 & 方法, 性能优化, 编程语言, 框架, 微服务, 芯片&算力

InfoQ 2022 年趋势报告：架构与设计篇

本文概要介绍了InfoQ编辑团队眼中2022年软件架构与设计主题的发展情况，重点介绍了现如今架构师的主要工作。

🔗架构, 语言 & 开发, 技术选型, 方法论, 操作系统, 编程语言, 微服务, 医疗, 企业动态, 行业深度

架构师成长路线

想一两句话把什么是架构师讲清楚，是非常困难的一件事。因为架构师这个角色是致力于解决高度复杂抽象的问题...

2022-05-05

发现更多内容



促进软件开发及相关领域知识与创新的传播

关于我们  
我要投稿  
合作伙伴  
加入我们  
关注我们

联系我们

内容投稿：editors@geekbang.com  
业务合作：hezuo@geekbang.com  
反馈投诉：feedback@geekbang.com  
加入我们：zhaopin@geekbang.com  
联系电话：010-64738142  
地址：北京市朝阳区望京北路9号2幢7层A701

InfoQ 近期会议

北京 · QCon 全球软件开发大会 2025.4.10-12  
上海 · AICon 全球人工智能开发与应用大会 2025.5.23-24  
北京 · AICon 全球人工智能开发与应用大会 2025.6.27-28

全球 InfoQ

🇨🇦 InfoQ En  
🇯🇵 InfoQ Jp  
🇫🇷 InfoQ Fr  
🇧🇷 InfoQ Br