

Homework: 阅读《软件架构师应该知道的 97 件事》，试列举出你认为 10 件重要的事情，并简要说明为什么？

答：

1. 偿还技术债务

在软件开发过程中，技术债务是一个不可避免的问题。它通常由于在项目初期为了追求快速交付而做出的短视决策，或是在项目进展中为了解决眼前问题而采取的权宜之计，逐渐积累而成。这些决策可能在当时看似合理，但长期来看，会增加系统的复杂度，降低其可维护性，甚至影响系统的性能和稳定性，从而对项目的可持续发展构成威胁。技术债务的存在，使得任何新的功能添加或者现有功能的修改都变得更加困难和耗时，因为开发人员需要在一个脆弱和复杂的代码基础上工作，这会降低开发效率，增加 bug 的风险。此外，长期面对一个难以维护和扩展的系统，开发人员的工作满意度会降低，进而影响团队的生产力。

因此，作为软件架构师，重视并管理技术债务是十分关键的。首先，架构师需要通过代码审查、自动化测试等措施，从一开始就尽可能减少技术债务的产生。其次，架构师应该定期评估项目中的技术债务，并将其管理作为项目计划的一部分。这包括为还债安排时间和资源，比如重构代码、更新过时的库和框架、优化数据库设计等，以确保技术债务得到有效控制。此外，架构师还需要在团队中培养对技术债务的认识，确保团队成员理解技术债务的长期影响，并鼓励他们在日常开发中采取负责任的做法，避免不必要的债务累积。

2. 分析客户需求背后的意义

埃纳尔·兰德雷（Einar Landre）在提倡分析客户需求背后的意义时，强调了深入理解用户真正需求的重要性。这一点至关重要，因为在软件开发中，表面的需求往往只是问题的一部分。真正理解需求背后的业务目标、用户的痛点和期望，可以帮助软件架构师和开发团队设计出更加贴合实际、更具有前瞻性和可持续发展能力的解决方案。

通过深挖需求背后的意义，团队可以避免过度设计或是开发出与用户实际需求偏离的功能，从而节省开发资源，提高项目的成功率。此外，这种做法还能增强产品的用户体验，因为它确保了产品功能紧密围绕用户的实际需求设计。最终，这种对需求深层次理解的方法，不仅提升了软件的商业价值，还能在竞争激烈的市场中为产品赢得更多的用户和更高的满意度。

3. 我们常常忽略了自己正在谈判

在多数情况下，个人或团队在进行项目讨论、需求分析或是解决方案设计时，可能会过分关注技术细节或立即解决问题，而忽视了谈判过程本身对于结果的重要性。

认识到谈判是一个互动过程，意味着需要了解对方的需求和限制，同时明确地表达自己的目标和约束。这不仅有助于达成更加平衡和双赢的结果，还能促进更好的团队合作和项目管理。通过积极参与谈判，我们可以更有效地管理期望，避免未来的误解和冲突，最终实现项目目标。

4. 一行代码比五百行架构说明更有价值

在软件开发中，一行实际运行的代码往往能够比长篇的架构说明文档更直观、更有效地传达一个设计理念或解决方案。代码是具体实现的体现，它不仅能够展示功能如何工作，还能够揭示实现细节和设计决策的考虑，这些都是架构说明文档难以完全传达的。

此外，实际的代码可以被立即验证和测试，这意味着它可以快速地展示想法的可

行性，同时为进一步的迭代和优化提供基础。在快速变化的软件开发领域，能够迅速实现和测试新的想法是至关重要的，因为它可以加速学习过程，帮助团队发现并修正问题，优化设计。

5. 打造数据库堡垒

打造数据库堡垒意味着采取一系列措施，确保数据库的安全性、稳定性和高效性。这一点至关重要，因为数据库通常包含了企业的核心数据和敏感信息，它的安全性直接关系到企业的运营安全 and 信息安全。此外，打造数据库堡垒还包括对数据库进行性能优化，确保数据库能够高效处理查询和事务，支持业务的正常运行。通过优化索引、调整查询语句等方式，提升数据库的处理能力和响应速度，从而提高整个系统的性能和用户体验。

6. 先尝试后决策

“先尝试后决策”强调了在面对复杂问题或新技术时，通过实际的试验或原型开发来获取真实反馈和数据，从而指导最终的决策制定。这种方法尤其适用于软件开发和技术领域，其中很多情况下理论分析和预测都有其局限性，而实际操作和测试能提供更直接和准确的信息。通过在较小的规模上先行尝试，还可以在全面推广前发现潜在的问题和风险，避免大规模实施时可能遭遇的失败和损失。

7. 控制项目规模

控制项目规模是指在项目管理过程中，合理设定和维护项目的边界，包括功能范围、资源分配、时间计划和成本预算，以确保项目目标的实现既符合预期也是可行的。这一点至关重要，因为项目规模的膨胀是导致项目延期、超预算、质量不达标甚至失败的主要原因之一。控制项目规模有助于维护项目目标和期望的一致性，减少变更和修改的需求，从而提高项目的成功率。通过控制项目规模，项目管理者可以有效地控制项目的执行过程，确保项目按时、按预算、按质量完成。

8. 仔细观察，别试图控制一切

在面对复杂的项目和团队管理任务时，应该采取更加灵活和开放的心态。通过细致的观察和理解团队的工作方式、项目进展以及外部环境的变化，我们可以更好地把握项目的方向和动态，而不是试图对每一个细节进行精确控制。这种方法允许更多的空间给团队自我组织和解决问题，激发创新和主动性。同时，它也帮助项目管理者从更宏观的视角审视项目，从而做出更加明智和有远见的决策。在这个过程中，项目管理者角色更多地转变为指导和支持，而非直接干预，这样不仅能够增强团队的凝聚力和动力，还能有效应对不断变化的环境和挑战。

9. 确保简单问题有简单的解

在面对问题时，应该首先寻找最直接、最简洁的解决方案，而不是过度设计或引入不必要的复杂性。这种方法有几个显著的好处：首先，简单的解决方案通常更容易理解和实施，这有助于减少开发时间和成本。其次，简单的解决方案也更易于维护和调试，因为其结构清晰，相关联的问题也更少。此外，简单的方案往往更加稳定可靠，因为复杂性本身就是软件错误的一个主要来源。通过追求简洁的解决方案，我们可以提高软件质量，提升用户体验，同时也促进了项目团队的效率。

10. 你不能不了解硬件

尽管软件的灵活性和抽象层次使得开发者可以在不深入了解底层硬件的情况下构建应用，但对硬件的深入理解无疑会带来更优的性能、更高的效率，以及更好的用户体验。这是因为软件最终是运行在硬件上的，硬件的特性和限制直接影响软件的性能和效能。此外，随着物联网、边缘计算等技术的兴起，软硬件结合越来越紧密，软件解决方案越来越需要考虑到硬件的特性和限制。因此，对硬件有一定的了解和认识，

对于软件开发来说是一个不可或缺的技能，它能够让开发者在设计 and 实现软件解决方案时更加游刃有余，提升整体的工作效率和项目的成功率。