1. **客户需求重于个人简历**  
   客户需求是项目成功的关键，架构师应优先考虑客户的实际需求，而非仅为了自己的简历或展示新技术。过于追求炫技往往会导致项目失败。
2. **简化根本复杂性，消除偶发复杂性**  
   解决根本问题时要避免引入不必要的复杂性。架构师的责任是识别和解决最根本的挑战，同时减少额外的技术复杂性。
3. **架构决定性能**  
   软件的性能和可扩展性在架构设计阶段就已经确定。后期的优化和调整无法替代一个科学的架构设计，必须从根本上进行合理设计。
4. **关键问题可能不是出在技术上**  
   成功的项目依赖于优秀的人才和团队的合作。架构师需要懂得管理团队并培养信任，技术仅是其中的一部分。
5. **以沟通为中心，坚持简明清晰的表达方式和开明的领导风格**  
   良好的沟通能有效推动项目进展。架构师应以简明清晰的方式表达自己的想法，并保持开放的领导风格，以便更好地与团队协作。
6. **故障终究会发生**  
   系统中不可避免地会发生故障，提前设计预防措施和容错机制是确保系统稳定性的关键。
7. **量化需求**  
   客户的需求必须量化，避免模糊的描述影响开发进度。通过具体的量化标准，可以确保需求的清晰度和可执行性。
8. **架构设计要平衡兼顾多方需求**  
   架构设计需要在技术需求和业务需求之间找到平衡。兼顾各方的需求，避免偏向某一方而忽视其他方面。
9. **程序设计是一种设计**  
   软件开发是一个设计与生产并重的过程。架构师应重视系统设计，而不仅仅关注具体的代码实现。
10. **不断学习和分享经验**  
    软件架构师需要不断学习新技术，分享知识和经验。通过分享和讨论，架构师能不断提升自己的技能，也能帮助团队成长。