



链滴

# 架构师之路一 - 架构师入门指引

作者: [jianzh5](#)

原文链接: <https://ld246.com/article/1583934732832>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

导读：本系列文章教你怎么样成为一名架构师，而本篇文章则带你先认识一下什么是架构师，架构师工作是什么？

## 为什么需要架构师

为什么需要架构师或者说架构师能解决什么样的问题，我们不妨先从两个不同的视角来看一下。

### 技术高手的视角

小张作为一名拥有3-5年开发经验的技术高手，他经常会思考以下几个问题：

- 我已经工作好几年了，将来如何发展？是要一直写代码吗？
- 是不是要往上走就得做管理？
- 在中国35岁之后不能再做技术了吗？
- 继续做技术是不是待遇上不如做管理？
- 如果继续做技术我还要学习什么？
- 如果改做管理我应该如何转型？
- 我适合做技术还是做管理，还是别的什么？

### 软件企业的视角

软件企业在的产品开发过程中也经常会思考以下几个问题：

- 为什么我们的产品交付周期为什么需要那么长时间？竞争对手可能只要半年，为什么我们需要1年？
- 为什么我们的产品总有这样那样的质量问题？程序员在开发的时候为什么不好好把控质量，上完线这样那样的问题？
- 为什么当初这个产品会选择这样的技术路线，技术选型的时候为什么不慎重？导致现在要用另一种术推翻重做，带来巨大的人力成本？
- 网站的用户越来越多，性能非常吃紧，想扩展却很难？
- 为什么这个产品的代码这么难维护，找谁改都说不敢动？
- 究竟谁能在技术上保证我们的产品或项目取得成功？

从不同的角度出发会引发出一连串的疑惑，那么能解决以上疑惑的角色就是**系统架构师**，也可以说我们需要**系统架构师**来解决这些问题。

## 架构、架构设计与架构师的相关概念

### 架构

架构，又名软件架构，也称为软件体系结构。软件架构就是一个系统的蓝图，是一种设计方案，将客的不同需求抽象成为抽象组件，并且能够描述这些抽象组件之间的通信和调用，及包括一些内部的关机制。

它有下面三个关键概念：

- 组件

通常是指开发或部署的一个单元，根据考查对象的大小，组件的粒度也有所区别。在做架构的时候我

需要把握好这个力度，不能陷入代码细节，如果过度的关注代码层面的力度，那对系统的整体把握可会出问题。

- 组件与组件之间的关系

是架构要考虑的重要因素，来自系统外部的请求通常是由多个组件协作完成的，系统内部结构是否良，很大程度上取决于组件之间的关系。

- 关键机制

是指影响到系统可用性、安全性、性能等重要非功能特性的一些技术方案，比如技术选型、关键设计处理流程等等。

## 系统架构 vs 架构设计

**系统架构** 是指系统在运行期的实际内部结构，架构设计是对这种内部结构的书面描述。

**架构设计** 是以需求分析为输入，通过架构师的分析，产出架构设计资料，用于指导后续概要设计、详设计、开发、测试、部署、上线运行。所以说如果架构设计做的不好或者没做架构设计，那么后面环的开发测试部署可能会出各种各样的问题。

## 架构设计 vs 概要设计

架构设计是以组件的视角来思考系统如何分解，并定义分解出来的组件之间的关系。

概要设计是从功能模块的视角来对系统进行分解，并定义这些功能模块之间的关系。现在提的比较少一般做完架构设计直接做详细设计即可！

以用户登录为例，从架构设计的角度可能就只是一个用户服务组件，而从概要设计的角度可能就是前页面、用户接口、数据库等一系列功能的设计。

## 架构师

架构师是负责系统架构的人、团队或组织，架构师是团队技术领导，从技术角度，承担项目技术的成或失败的责任。在其领域内能够全局把握的人，能够给出其负责范围内的总体设计，并能解决关键问、指导其他人员落实设计。

往往后端开发出身的架构师对后台开发这一块有很丰富的相关经验，但是还需要对前端，APP端、测、部署等领域也需要了解掌握，需要能做到掌控全局，这也是成为架构师需要去修炼的地方。

**注解：架构师在一个团队中的权利很大，在技术上大家都要听你的，但是同时你也要承担相应的义务一旦项目由于技术原因失败，那你就是第一责任人**

## 架构师的价值

李智慧老师在《大型网站技术架构 核心原理与案例分析》说过软件架构师的最大价值不在于掌握多少进的技术，而在于具有将一个系统切分成N个低耦合的子模块的能力，这些子模块包含横向的业务块，也包含纵向的基础技术模块。这种能力一部分源自专业的技术和经验，还有一部分源自于架构师业务场景的理解、对人性的把握、甚至对世界的认知。”



上面这张图表示未经过架构设计的系统，大家想怎么建就怎么建，用过几年后系统之间的关系没人能清楚，自然程序员不敢随便改其中的代码。



而经过良好的架构设计后系统之间逻辑清晰，可以很容易进行扩展。

## 架构、架构师、架构设计之间的关系

下面一张图很容易看出架构、架构师以及架构设计之间的关系



## 架构师能力模型

作为架构师需要拥有以下12个能力模型：



### ● 沟通协作：

架构师需要经常跟产品经理、项目经理甚至客户打交道，所以沟通能力对架构师来说非常重要，能力结如下

① 具备优秀的口头、书面及表达技巧

- ② 优先的聆听者和观察者
- ③ 传达和激发团队，共享架构，确保达成一致
- ④ 个人品牌，值得信任
- ⑤ 推动良好的团队协作，合作共赢

- 自我驱动：

架构师为什么能够成为架构师？因为他们都会有强大的自我驱动力，总结如下

- ① 积极主动，承担职责以外的事情
- ② 持之以恒，长期保持
- ③ 严格要求自己，不满足现状

- 高效学习：

这个能力所有做开发的都需要具备

- ① 发现自身知识结构的优劣
- ② 形成自己的学习模式
- ③ 目标导向，学习目标要明确
- ④ 学习需要反复强化，不断实践运用

- 良好心态：

- ① 开放心态，能够取长补短，要多与分歧者沟通
- ② 责任心，敢于决策，为决策结果负责
- ③ 严于律己，宽以待人，积极向上

- 识别问题：

公司花钱聘请你来的目的是让你来解决问题，而解决问题的前提是先识别问题，而架构师需要快速准的识别问题，主要分为以下几个方面

- ① 识别问题以及问题的主体，把问题本身先搞清楚
- ② 发现问题永远比解决问题更加重要
- ③ 可以通过利益者全面沟通、竞争对手分析等手段来识别问题
- ④ 问题的优先级，可以用钱或者对业务的影响面来衡量

- 抽象思维：

作为架构师这个能力尤其重要

- ① 能够分解出共性和个性，提炼出共性
- ② 需求概念化（由实到虚总结升华）并归类（核心/非核心等），然后分而治之
- ③ 抽象的前提是对个性的深入理解

- 认识深度

- ① 深层次挖掘（由虚到实）问题的本质
- ② 技术的本质
- ③ 业务的本质

- ④ 利益相关者的本质

- 平衡取舍

这个能力也非常重要，毕竟公司给你资源是有限的。如果给你无限的资源，那就不需要做架构了，架构师就是需要在有限的资源中最大化经济效益。往往做架构设计就是一个取舍的过程。

- ① 利益者之间利益程度的的平衡取舍
- ② 确保架构在现有有限资源约束下最合理，最大化经济效益
- 业务能力

不了解业务肯定做不出良好的架构设计的，需要了解业务的现状以及未来的发展趋势。

- ① 对于所在业务和领域要有较深的理解
- ② 能够对业务需求进行分解和未来判断
- ③ 好的架构师也是好的产品经理
- 技术能力

这是作为架构师最基本的能力

- ① 具备编码/设计/攻关等能力，丰富项目经验
- ② 技术深度，某一个领域的技术专家
- ③ 技术广度，技术知识面比较广
- ④ 技术高度，技术前瞻和判断力，技术支撑和引导业务
- 想象力
- ① 技术创新，以业务为中心的方式识别、评估和注入颠覆性新技术的能力
- ② 战略性规划，能够为实现潜在目标设计战略路线图并推动落地
- ③ 企业执行，企业精神、承担逾期风险、交付成果
- 架构方法论
- ① 多学习掌握业内/公司成熟的方法论，并且实践体会
- ② 自己结合项目循环总结，形成自身的架构方法论体系

## 架构师修炼方法

架构师可以从以下几个角度进行自我修炼

- 丰富实战
- 1、先在一个产品/项目做的比较深入，然后考虑多产品/项目的实践；
- 2、积极主动进行可复用模块提炼以及思路和手段的改进，减少无效重复实践
- 3、在完成本职工作的前提下，增加影响力在更大范围实践
- 深度思考
- 1、六步思考：确定与定义问题、分析问题、寻找解决问题的方法、做出决定、采取行动、评估结果控制
- 2、总结思考，形成自己的知识经验财富
- 融入圈子
- 1、融入到部门/公司架构师的圈子，尤其是要找到自己心中的导师；
- 2、融入行业相同的技术圈子，互相学习交流
- 3、经常写博客、参与开源社区、演讲以及培训等手段

- 不断学习

- 1、系统化知识体系的学习，权威书籍/网站/微信公众号等
- 2、新技术的感知、运用、分析以及场景运用
- 3、参加各种培训、分享以及交流等，与专家讲师碰撞学习

## 架构师成长路径

架构师的前身是一名中高级开发人员，他们通常会具备以下几个特征：

- 工作三五年，精通一两种编程语言；
- 精通几个框架，如SSH；
- 能够搭建项目的代码框架，开发核心模块，组织共通类库，编写示例程序；
- 能够解决一些开发过程中的难题；
- 能够对下级程序员进行指导；
- 能够负责一些中小模块的设计；
- 知识和能力体系与其承担什么项目有很大相关性；

在职业发展中他们有几条路径可走



走管理路线可以成长为项目经理、部门经理

走技术路线可以成为某方面的技术专家、架构师、CTO

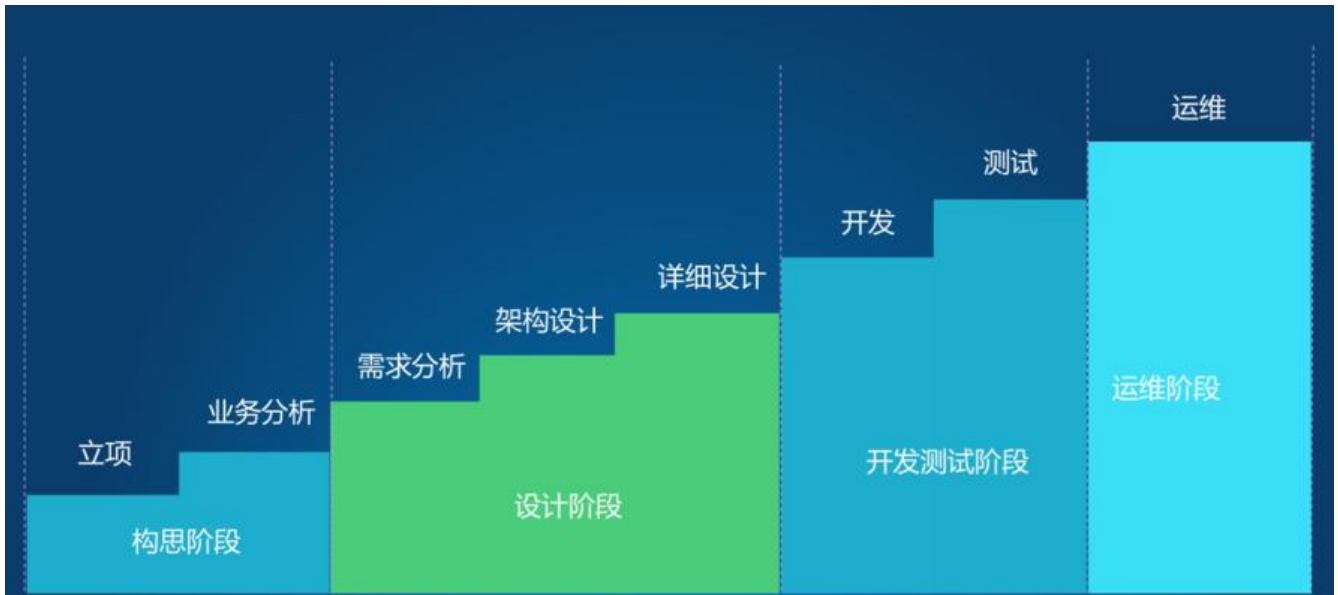
**成为架构师** 意味着需要具备更高的能力，并承担更大的责任。

## 架构师工作指南

### 工作职责

在标准软件研发流程体系中，软件研发分为构思阶段、设计阶段、开发测试阶段，运维阶段。而架构需要参与整个开发流程的生命周期。





我们接下来看看架构师在每个阶段需要干什么事。

- 立项阶段的职责（主要是向咨询或需求分析人员提供技术咨询）
  - 进行总体架构设想
  - 论证技术可行性
  - 验证某些关键技术问题

- 业务分析和需求分析阶段的职责

协助业务分析人员产出业务分析成果，包括以下事项：

- 与产品团队协作完成业务分析文档
- 参与系统业务价值讨论

协助需求分析人员完成需求分析，包括以下事项：

- 对产品团队进行技术支撑，解答产品团队的技术疑问
- 把握产品团队的需求成果，确保形式和内容符合架构设计输入需要，确保功能可实现，非功能需求指标合理，成本和工期可接受
- 完善需求分析

- 架构设计阶段的职责（独立完成架构设计）

- 逻辑架构设计，将系统分解为非技术性的逻辑组件，并定义其间的关系
  - 物理架构设计，将逻辑架构中的组件和关系进行技术化、具体化
  - 对于没有经验的技术点，验证其可行性
  - 性能验证
  - 技术选型时对多种方案对比验证
  - 架构评审，设计完成时邀请其他成员、组外专家、领导、高阶架构师对自己的工作成果进行评审
  - 软硬件采购申请，对设计、开发、测试、部署各环节需要的硬件及软件编写采购清单，提交申请
- 概要设计和详细设计阶段的职责（与开发组长一起完成概要设计）

- 功能清单整理，根据系统用例和架构设计中的组件定义推导出功能清单

- 接口定义，包括组件间的通信机制定义和功能模块间的接口定义

与开发组长一同确定详细设计的范围，指导中级开发人员完成必要的详细设计

- 初期指导，说明架构设计意图、详细设计注意事项
- 设计检查与评审，确保详细设计符合架构设计要求
- 参与数据库设计，确保数据库设计符合架构设计要求，主要考虑性能、数据量等问题
- 参加界面设计评审

- 开发阶段的职责

指导开发人员落实架构设计中要开发组件的实现，包括以下事项：

初期指导：

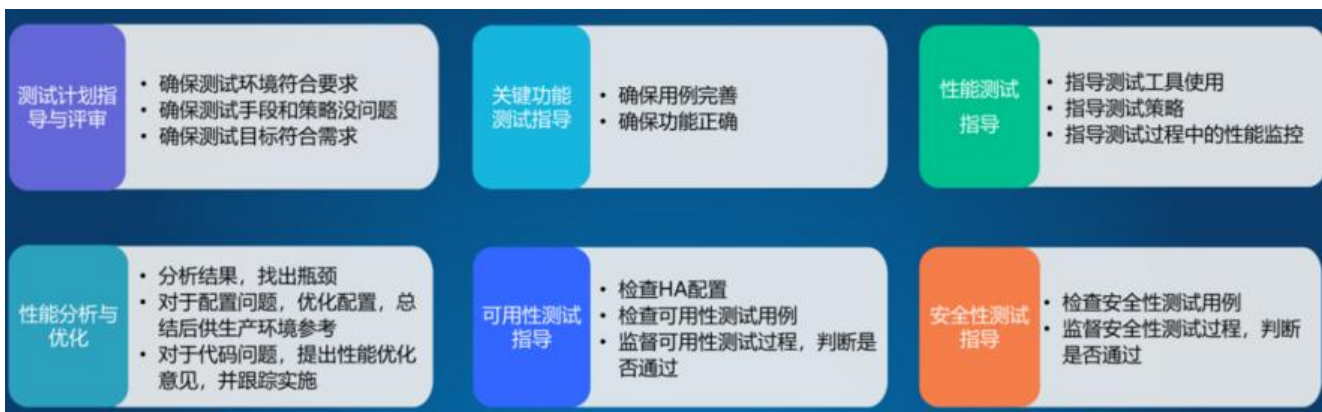
- 向开发团队说明开发相关的架构设计意图
- 配合开发组长搭建开发环境，建立各组件的代码工程
- 解答开发团队的疑问

代码检查与评审：

- 检查工程结构是否合理
- 检查组件的版本是否合适
- 检查接口是否与架构设计一致
- 检查主要处理流程的调用关系
- 检查关键功能的实现
- 检查通信方式
- 检查并发处理方式
- 检查连接池、线程池等资源的利用
- 检查缓存的实现方式和策略
- 检查配置项实现方式
- 检查构建脚本

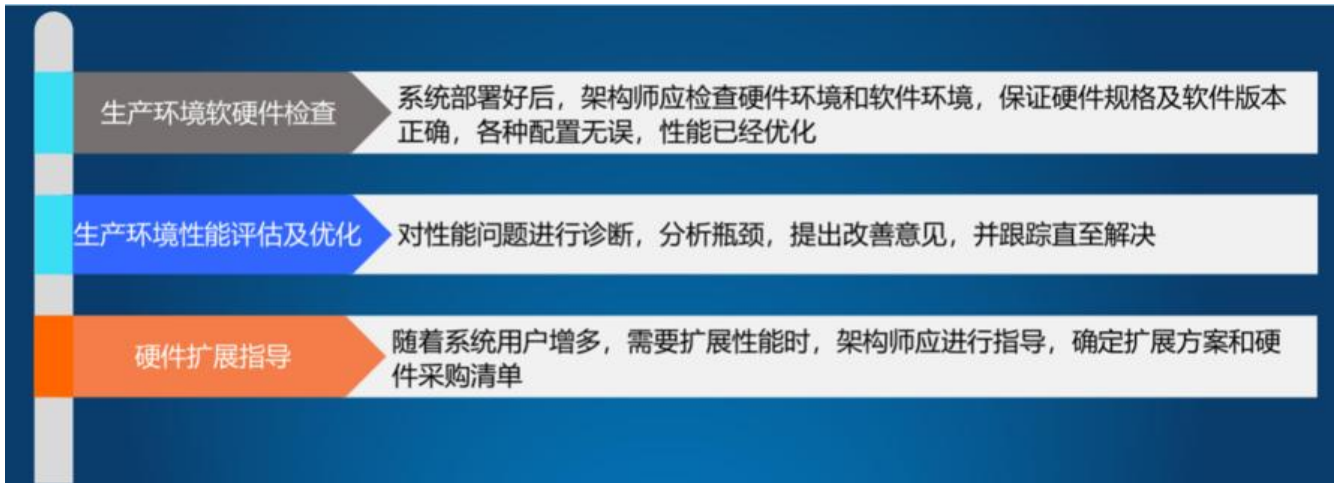
- 测试阶段的职责

指导测试人员检验架构设计中非功能特性的实现，包括以下事项：



- 运维阶段的职责

指导运维人员部署系统以及在后续运维过程中进行指导，包括以下事项：



- 架构师在组织中的职责

架构师是高级技术人员，在项目之外，还需要承担一定的组织建设职责，包括以下事项：



## 工作流程

架构师在项目中的活动需要有一定的流程，正常过程如下：

- 制定项目的架构工作计划
- 完善需求分析
- 进行架构设计
- 指导概要设计
- 指导详细设计
- 指导开发
- 指导测试
- 指导上线运维

- 管理架构变更

## 周边协作

架构师由于需要参与整个项目的生命周期，所以基本需要与所有相关人员进行协作，具体可参看下图：



## 资源保障

架构师在工作过程中会有一些资源需求，可通过以下方式进行获取：



## 架构师的考核

可以通过以下维度对架构师进行综合考核：

- 考核架构工作计划执行的完整性
- 考核架构设计文档的质量
- 考核指导、检查和评审的效果
- 考核项目非功能性需求的满足情况

- 考核架构师知识经验的分享情况
- 考核架构师对公司产品的改进情况