

# 新入职一家公司如何快速进入工作状态？

朋友投稿的一篇文章！强烈建议每一位即将入职/在职的小伙伴看看这篇文章，看完之后可以帮助你少踩很多坑。整篇文章逻辑清晰，内容全面！

原文：<https://www.cnblogs.com/hunternet/p/14675348.html>

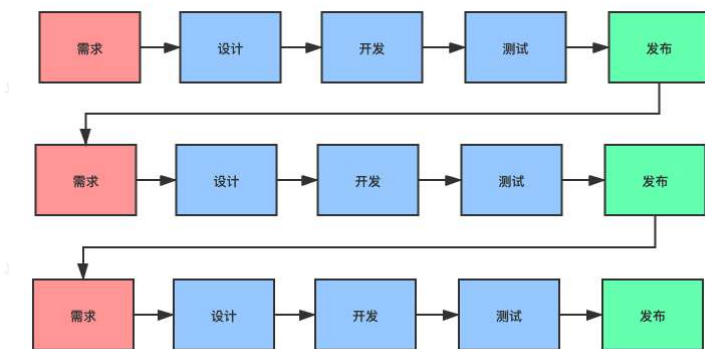
每到一个新的公司面临的的可能都是新的业务、新的技术、新的团队.....这些可能会打破你原来工作思维、编码习惯、合作方式.....

而于公司而言，又不能给你几个月的时间去慢慢的熟悉。这个时候，如何快速进入工作状态，尽快发挥自己的价值是非常重要的。

有些人可能会很幸运，入职的公司会有完善的流程与机制，通过一带一、各种培训等方式可以在短时间内快速的让新人进入工作状态。有些人可能就没有那么幸运了，比如如我在几年前跳槽进入某厂的时候，当时还没有像我们现在这么完善的带新人融入的机制，又赶上团队最忙的一段时间，刚一入职的当天下午就让给了我几个线上问题去排查，也没有任何的文档和培训。遇到情况，很多人可能会因为难以快速适应，最终承受不起压力而萌生退意。

那么，我们应该如何去快速的让自己进入工作状态，适应新的工作节奏呢？

新的工作面对着一堆的代码仓库，很多人常常感觉无从下手。但回顾一下自己过往的工作与项目的经验，我们可以发现它们有着异曲同工之处。当开始一个新的项目，一般会经历几个步骤：需求->设计->开发->测试->发布，就这么循环往复，我们完成了一个又一个的项目。



而在这个过程中主要有四个方面的知识那就是业务、技术、项目与团队贯穿始终。新入职一家公司，我们第一阶段的目标就是要具备能够跟着团队做项目的能力，因此我们所应尽快掌握的知识点也要从这四个方面入手。

## 业务

很多人可能会认为作为一个技术人，最应该了解的不应该是技术吗？于是他们在进入一家公司后，就迫不及待的研究起来了一些技术文档，系统架构，甚至抱起来源代码就开始“啃”，如果你也是这么做的，那就大错特错了！在几乎所有的公司里，技术都是作为一个工具存在的，虽然它很重要，但是它也是为了承载业务所存在的，技术解决了如何做的问题，而业务却告诉我们，做什么，为什么做。一旦脱离了业务，那么技术的存在将毫无意义。

想要了解业务，有两个非常重要的方式

### 一是靠问

如果你加入的团队，有着完善的业务培训机制，详尽的需求文档，也许你不需要过多的询问就可以了解业务，但这只是理想中的情况，大多数公司是没有这个条件的。因此我们只能靠问。

这里不得不提的是，作为一个新人一定要有一定的脸皮厚度，不懂就要问。我见过很多新人会因为内向、腼腆，遇到疑问总是不好意思去问，这导致他们很长一段时间都难以融入团队、承担更重要的责任。不怕要怕挨训、怕被骂，而且我相信绝对多数的程序员还是很好沟通的！

### 二是靠测试

我认为测试绝对是一个快速了解团队业务的方式。通过测试我们可以走一走自己团队所负责项目的整体流程，如果遇到自己走不下去或想不通的地方及时去问，在这个过程中我们自然而然的就可以快速的了解到核心的业务流程。

在了解业务的过程中，我们应该注意的是不要让自己过多的去追求细节，我们的目的是先能够整体了解业务流程，我们面向哪些用户，提供了哪些服务.....

## 技术

在我们初步了解完业务之后，就该到技术了，也许你已经按捺不住翻开源代码的准备了，但还是要先提醒你一句先不要着急。

这个时候我们应该先按照自己了解到的业务，结合自己过往的工作经验去思考一下如果是自己去实现这个系统，应该如何去做？这一步很重要，它可以在后面我们具体去了解系统的技术实现的时候去对比一下与自己的实现思路有哪些差异，为什么会有这些差异，哪些更好，哪些不好，对于不好我们可以提出自己的意见，对于更好的我们可以吸收学习为己用！

接下来，我们就是要了解技术了，但也不是一上来就去翻源代码。**应该按照从宏观到细节，由外而内逐步地对系统进行分析。**

首先，我们应该简单的了解一下 **自己团队/项目的所用到的技术栈**，Java 还是.NET、亦或是多种语言并存，项目是前后端分离还是服务端全包，使用的数据库是 MySQL 还是 PostgreSQL.....，这样我们可能会对所用到的技术和框架，以及自己所负责的内容有一定的预期，这一点有的人可能在面试的时候就会简单了解过。

下一步，我们应该了解的是 **系统的宏观业务架构**。自己的团队主要负责哪些系统，每个系统又主要包含哪些模块，又与哪些外部系统进行交互.....对于这些，最好可以通过流程图或者思维导图等方式整理出来。

然后，我们要做的是看一下 **自己的团队提供了哪些对外的接口或者服务**。每个接口和服务所提供功能是什么。这一点我们可以继续去测试自己的系统，这个时候我们要看一看主要流程中主要包含了哪些页面，每个页面又调用了后端的哪些接口，每个后端接口又对应着哪个代码仓库。(如果是单纯做后端服务的，可以看一下我们提供了哪些服务，又有哪些上游服务，每个上游服务调用自己团队的哪些服务.....)，同样我们应该用画图的形式整理出来。

接着，我们要了解一下 **自己的系统或服务又依赖了哪些外部服务**，也就是说需要哪些外部系统的支持，这些服务也许是团队之外、公司之外，也可能是其他公司提供的。这个时候我们可以简单的进入代码看一下与外部系统的交互是怎么做的，包括通讯框架(REST、RPC)、通讯协议.....

到了代码层面，我们首先应该了解每个模块代码的层次结构，一个模块分了多少层，每个层次的职责是什么，了解了这个就对系统的整个设计有了初步的概念，紧接着就是代码的目录结构、配置文件的位置。

最后，我们可以寻找一个示例，可以是一个接口，一个页面，让我们的思路跟随者代码的运行的路线，从入参到出参，完整的走一遍来验证一下我们之前的了解。

到了这里我们对于技术层面的了解就可以先告一段落了，我们的目的知识对系统有一个初步的认知，更细节的东西，后面我们会有大把的时间去了解

## 项目与团队

上面我们提到，新入职一家公司，第一阶段的目标是有跟着团队做项目的能力，接下来我们要了解的就是项目是如何运作的。

我们应该把握从需求设计到代码编写入库最终到发布上线的整个过程中的一些关键点。例如项目采用敏捷还是瀑布的模式，一个迭代周期是多长，需求的来源以及展现形式，有没有需求评审，代码的编写规范是什么，编写完成后如何构建，如何入库，有没有提交规范，如何交付测试，发布前的准备是什么，发布工具如何使用.....

关于项目我们只需要观察同事，或者自己亲身经历一个迭代的开发，就能够大概了解清楚。

在了解项目运作的同时，我们还应该去了解团队，同样我们应该先从外部开始，我们对接了哪些外部团队，比如需求从哪里来，是否对接公司外部的团队，提供服务的上游团队有哪些，依赖的下游团队有哪些，团队之间如何沟通，常用的沟通方式是什么.....

接下来则是团队内部，团队中有哪些角色，每个人的职责是什么，这样遇到问题我们也可以清楚的找到对应的同事寻求帮助。是否有一些定期的活动与会议，例如每日站会、周例会，是否有一些约定俗成的规矩，是否有一些内部评审，分享机制.....

## 总结

新入职一家公司，面临新的工作挑战，能够尽快进入工作状态，实现自己的价值，将会给你带来一个好的开始。

作为一个程序员，能够尽快进入工作状态，意味着我们首先应该具备跟着团队做项目的能力，这里我站在了一个后端开发的角度上从业务、技术、项目与团队四个方面总结了一些方法和经验。

关于如何快速进入工作状态，如果你有好的方法与建议，欢迎在评论区留言。

最后我们用一张思维导图来回顾一下这篇文章的内容。如果你觉得这篇文章对你有帮助，可以关注文末公众号，我会经常分享一些自己成长过程中的经验与心得，与大家一起学习与进步。