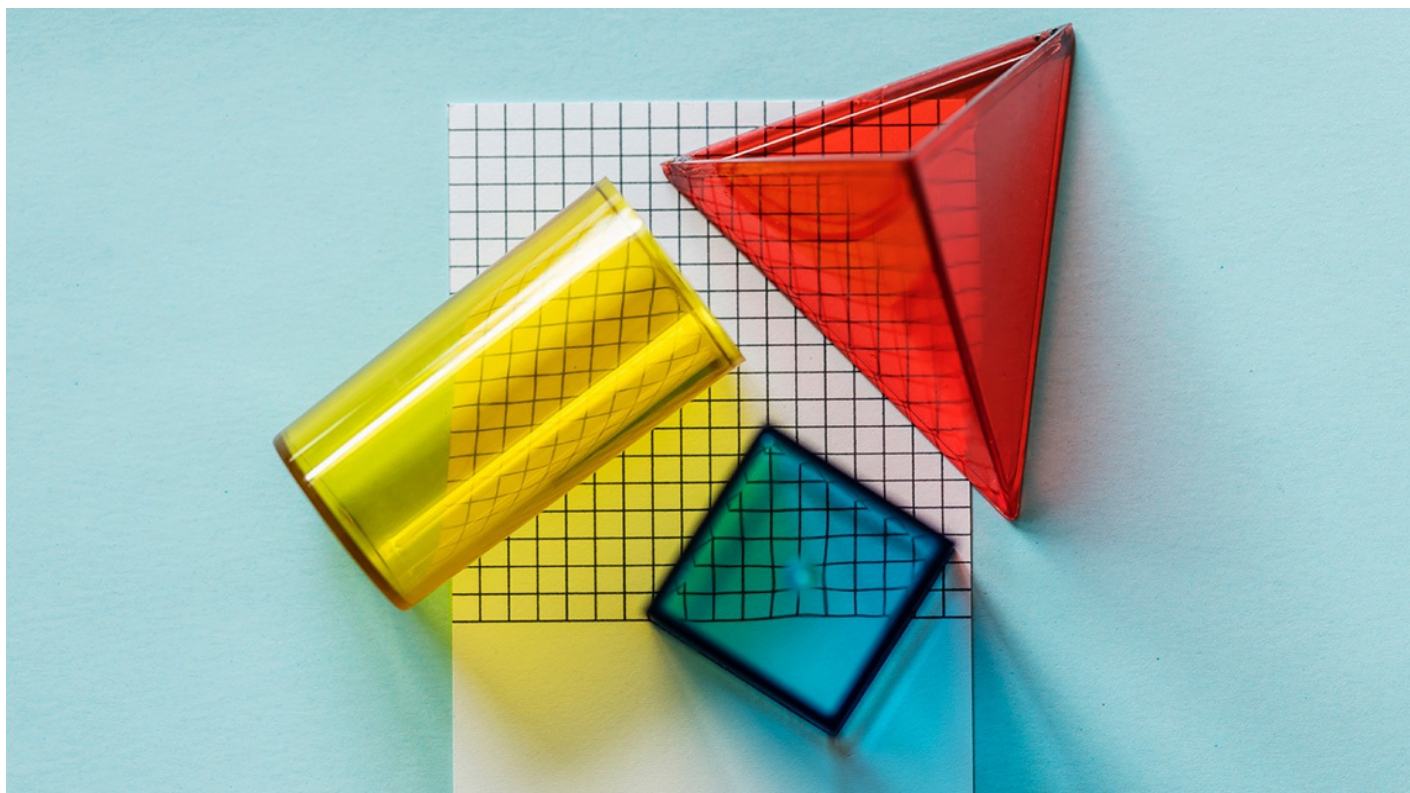


07讲解决了很多技术问题，为什么你依然在“坑”里



在前面的内容中，我给你介绍了几个体现“以终为始”原则的实践，包括怎样界定工作是否完成的 DoD、怎样判定需求是否完成的验收标准、还有怎样验证产品经理给出的产品特性是否合理的精益创业理念。

了解了这些内容，可能你会想：我为什么要关心这些啊？我是程序员啊！难道我不应该安安静静地写程序吗？为什么要操心其他人的工作做得好坏？如果我管了那么多事，我还是不是一个程序员，到底哪里才是我的“终”呢？

今天这一讲，我们就来聊聊这个让许多人困惑的问题。因为只有要跳出程序员的角色看问题，工作才会变得更加高效。

“独善其身”不是好事

在需要与人协作的今天，独善其身可不一定是好的做法。我先给你讲一个发生在我身边的故事。

有一次，我的团队要开发一个数据服务层，准备作为一个基础设施提供给核心业务系统。开发没多久，一个团队成员和我说，他的工作进展不顺利，卡在了一个重要问题上，他想不明白该如何在多个实例之间分配 ID。

我听完之后，有些疑惑，为什么要考虑这个和功能无关的问题呢？他解释说，因为我们的系统需要保证消息的连续性，所以他设计了消息 ID，这样下游系统可以通过消息 ID 识别出是否有消息丢失。

这是没错的，但我奇怪的是，他为什么要在多个实例之间协调呢？他给出的理由是，这么做，是出于考虑应对将来有多实例并发场景的出现。然而事实是，我们当下的需求应对的是单实例的情况。

我了解情况之后，马上跟他说清楚这一点，让他先把第一步做出来。这个同事还是有些担心未来如何做扩展。我告诉他，别纠结，先把第一步做出来，等后面真的有需求，我们再考虑。同事欣然答应了。

其实，这个同事的技术能力非常强，如果我不拦着他，他或许真能实现出一个完美的技术方案，但正如他自己所纠结的那样，这个方案可能要花掉他很长时间。但这真的是我们想要的吗？以现阶段的目标来看，根本没有这样的需求。

我们一直在强调“以终为始”。所谓“终”，其实就是我们的做事目标。虽然大家工作在一起，朝着一个共同的大目标前进，但真

的到了一个具体的问题上，每个人看到的目标却不尽相同。

我之所以能把同事从一个纠结的状态中拉出来，是因为我看到的是需求，而他看到的是一个要解决的技术问题。所以，我们俩在对目标的理解上是有根本差异的。

你也许会认为，我和同事之所有这样的差异，是角色上的差异，我在项目里承担的角色要重一些，而且我的工作时间比同事要长一些。但不知道你有没有想过，不同角色的差异到底在哪里呢？

角色的差异

作为一个在职场工作的人，每个人都有一颗渴望得到认可的心，希望自己在职业的阶梯上步步高升。假如今天就让你往上走一个台阶，比如，你原来在项目里打杂，现在成为项目的主力，或者，你已经对项目细节驾轻就熟，即将委任你为项目负责人。你是否能胜任呢？

你需要补充的东西是什么？换句话说，你和你职业台阶中的上一级那个人，差异到底是什么？

也许你会说，他比我来的时间长，或者说，他每天的主要工作就是开会。如果真的是这样，那是不是只要你凑足这个条件，就可以到达他的位置呢？显然不是。

不同角色工作上真正的差异是上下文的不同。

这是什么意思呢？以前面的问题为例，你在项目里打杂，你只能关注到一个具体的任务，而项目主力心目中是整个系统。**虽然写的代码都一样，但你看到的是树木，人家看到的是森林，他更能从全局思考。**

同样，项目负责人的工作，虽然包括在项目组内的协调，但还有一部分工作是跨项目组的，他需要考虑你们项目组与其他组的互动。所以，他工作的上下文是在各组之间，包括技术和产品等方面。

再上升一个层面，部门负责人要协调内部各个组，同时要考虑部门之间的协调。而公司负责人考虑的上下文甚至要跳脱公司内部，进入到行业层面。

你可能会问，好了，我知道不同角色的上下文有差异了，但这对我意味着什么呢？

我们先从工作角度看。回到前面我分享的那个故事，你可能注意到了，**我并不是靠技术能力解决了问题，而是凭借对需求的理解把这个问题绕过去了。**

之所以我能这样做，原因就在于我是在一个更大的上下文里工作。类似的故事在我的职业生涯中发生过无数次，许多令程序员愁眉不展的问题，换个角度可能都不是问题。

技术是一把利刃，程序员相信技术可以改变世界，但并不是所有问题都要用技术解决。有这样一种说法，手里有了锤子，眼里都是钉子。花大力气去解决一个可能并不是问题的问题，常常是很多程序员的盲区。

之所以称之为盲区，是因为很多人根本看不见它，而看不见的原因就在于上下文的缺失，也就是说，你只在程序员的维度看问题。

多问几个为什么，交流一下是不是可以换个做法，许多困惑可能就烟消云散了。**而能想到问这样的问题，前提就是要跳出程序员角色思维，扩大自己工作的上下文。**

虽然我不是项目主力，但不妨碍我去更深入地了解系统全貌；虽然我不是项目负责人，但不妨碍我去了解系统与其他组的接口；同样，虽然我不是项目经理，但我可以去了解一下项目经理是怎样管理项目的；虽然我不是产品经理，但了解一个产品的设计方法对我来说也是有帮助的。

当你对软件开发的全生命周期都有了认识之后，你看到的就不再是一个点了，而是一条线。与别人讨论问题的时候，你就会有

更多的底气，与那些只在一个点上思考的人相比，你就拥有了降维攻击的能力。

现在你知道为什么你的工作总能让老板挑出毛病了吧！没错，工作的上下文不同，看到的维度差异很大。单一维度的思考，在多维度思考者的眼里几乎就是漏洞百出的。

当扩大了自己工作的上下文时，我们的目标就不再局限于一个单点，而是会站在更高的维度去思考，解决问题还有没有更简单的方案。许多在低一级难以解决的问题，放到更大的上下文里，根本就不是问题。

我的职业生涯中经常遇到这样的情况，在一个特定的产品设计下，我总觉得设计的技术方案有些不优雅的地方，而只要产品设计微调一下，技术方案一下子就会得到大幅度提升。在这种情况下，我会先去问产品经理，是否可以这样调整。只要不是至关重要的地方，产品经理通常会答应我的要求。

在更大的上下文工作

扩展自己工作的上下文，目光不再局限于自己的一亩三分地，还可以为自己的职业发展做好布局。在这个方面，我给你分享一个不太成功的案例，就是我自己的故事。

我是属于愚钝型的程序员，工作最初的几年，一直把自己限定在程序员的上下文里，最喜欢的事就是安安静静地写代码，把一个系统运作机理弄清楚会让我兴奋很长一段时间。

我的转变始于一次机缘巧合，当时有一个咨询项目，负责这个项目的同事家里有些事，需要一个人来顶班，公司就把我派去了。

到了咨询项目中，我自己习惯的节奏完全乱掉了，因为那不是让代码正常运作就可以解决的问题，更重要的是与人打交道。

有很长一段时间，我一直处于很煎熬的状态，感谢客户没有把我从这个项目赶出来，让我有了“浴火重生”的机会。

为了让自己从这种煎熬的状态中摆脱出来，我必须从代码中走出来，尽量扩大自己思考的边界。经过一段时间的调整，我发现与人打交道也没那么难，我也能更好地理解一个项目运作的逻辑，因为项目运作本质上就是不同人之间的协作。

突破了自己只愿意思考技术的限制，世界一下子宽阔了许多。所以，后来才有机会更多地走到客户现场，看到更多公司的项目运作。虽然我工作过的公司数量并不多，但我却见过很多公司是如何工作的。

再后来，我有机会参与一个新的分公司建设工作中，这让我有了从公司层面进行思考的角度。对于员工招聘和培养，形成了自己一套独立的思考。

这些思考在我创业的过程中，帮我建立了一支很不错的团队。而创业的过程中，我又有了更多机会，去面对其他公司的商务人员，从而建立起一个更大的上下文，把思考从公司内部向外拓展了一些。

回过头来看自己的生涯时，我发现，因为不愿意拓展自己的上下文，我其实错过了很多职业发展的机会。所幸我还有机会突破自己，让自己走出来，虽然走的速度不如理想中快，但至少一直在前进，而不是原地打转。这也是我告诫你一定要不断扩大自己工作上下文的原因。

机会总是垂青那些有准备的人，尤其在公司规模不大的时候，总有一些跳跃式的发展机会。

我见过有人几年之内从程序员做到公司中国区负责人，只是因为起初公司规模不大，而他特别热心公司的很多事情，跳出了固定角色的思维。所以，当公司不断发展，需要有人站出来的时候，虽然没有人是完全合格的，但正是他的热心，让他有了更多的维度，才有机会站到了前排。

当然，随着公司规模越来越大，这种幅度极大的跳跃是不大可能的。江湖上流传着一个华为的故事，一个新员工给任正非写了封万言书，大谈公司发展，任正非回复：“此人如果有精神病，建议送医院治疗，如果没病，建议辞退。”

因为一旦公司规模大了，你很难了解更大的上下文，很多关于公司的事情，你甚至需要从新闻里才知道。

本质上，一个人能在自己的工作范围内多看到两三级都是有可能的。在公司规模不大时，从基层到老板没有太多层级，跳跃就显得很明显，而公司一大，层级一多，从低到顶的跳跃就不太可能了，但跨越级别跳跃是可能的。

所以希望你跳出程序员思维，这不仅仅是为了工作能够更高效，也是希望你有更好的发展机会。

总结时刻

总结一下今天的内容。程序员总喜欢用技术去解决一切问题，但很多令人寝食难安的问题其实根本不是问题。之所以找不出更简单的解决方案，很多时候原因在于程序员被自己的思考局限住了。

不同角色工作真正的差异在于上下文的差异。在一个局部上下文难以解决的问题，换到另外一个上下文甚至是可以不解决的。所以说无论单点有多努力也只是局部优化，很难达到最优的效果。

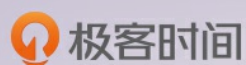
想把工作做好，就需要不断扩大自己工作的上下文，多了解一下别人的工作逻辑是什么样的，认识软件开发的全生命周期。

扩大自己的上下文，除了能对自己当前的工作效率提高有帮助，对自己的职业生涯也是有好处的。随着你看到的世界越来越宽广，得到的机会也就越来越多。

如果今天的内容你只记住一件事，那请记住：**扩大自己工作的上下文，别把自己局限在一个“程序员”的角色上。**

最后，我想请你分享一下，在你的工作中，有哪些因为你扩大了工作上下文而解决的问题呢？欢迎在留言区写下你的想法。

感谢阅读，如果你觉得这篇文章对你有帮助的话，也欢迎把它分享给你的朋友。



10x 程序员工作法

掌握主动权，忙到点子上

郑晔

火币网首席架构师
前 ThoughtWorks 首席咨询师
TGO 鲲鹏会会员



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

精选留言



王小勃
打卡

扩大工作上下文，跳出程序员思维

更新请加微信1182316662 众筹更多课程209

醍醐灌顶，受教！

2019-01-09 00:41



ZackZeng

之前一个项目，需要调用三方的API拿到客户购买历史，通过和上下游服务交涉，明白购买历史只对于会计计算有用，而会计允许一定范围误差，确认不调用三方API的结果在误差范围之内，结果就是节约了大概4s的API延迟

2019-01-09 06:48

作者回复

赞！

2019-01-09 08:49



Vackine

听了这么多第一次留言。还是研究生，马上毕业，听了之后，感觉对自己以后职业的发展有了更清楚的想法认识和计划。事物发展的过程都是由点到面不断往外扩充的，要想成长也是如此，从最开始看到的点，前后上下文的联系，以及思维的转变，才是成长更近一步的表现。老师讲的非常棒，跟下去之后对自己之后毕业入职一定会有很大的帮助！

2019-01-09 00:54



香烟情人

我在项目中也多次遇到过通过扩大上下文而解决问题的情况，比如有一次客户提了一个定期从我方系统提取数据同步给某外围系统的需求，开发过程中我们发现提数逻辑涉及关联一张上亿多数据的大表，查询性能很差，技术层面无论如何是无法解决的，最后我们从需求层面出发，着手绕过这个坎，经过分析，发现关联这张大表的作用主要就是为了过滤掉一些脏数据，再结合客户最终的使用场景分析，发现其实由外围系统进行此项过滤会更容易，最后我们说服了客户和友商，不仅很好的绕过了这个坎，也成功实现了该需求；类似的案例还有很多，技术解决不了的问题就通过非技术手段解决，这个案例中开发人员搞了几天一直没能解决，后来报了进度风险，最后版本经理想到从需求层面绕过去，这或许就是他们两个角色的上下文不同的区别吧

2019-01-09 23:36

作者回复

赞！

2019-01-10 08:09



此方彼方Francis

老板也曾经跟我聊过跳出程序员思维这个事情，会说到控制成本啊、具备一些产品思维啊这些方面，但是由于个人愚钝，总觉得这个概念很模糊，自己没什么比较明确的提升办法。

看到这篇文章让我有点豁然开朗的感觉，以后起码我可以从扩展上下文这个方面入手，在这个过程中应该能有不少收获。

感觉买的很值，感谢！

2019-01-10 09:29



大彬

我很喜欢上下不同这个描述。我曾经以为，具有主管的能力就够了，后来发现这还不够，缺少主管这个位置的视野和背景，就是上下文这个词。我们组的主管，是上级主管兼任的，我想，我是不是能够成为组的主管？有次主管有事没来，让我按他说的，给其他同事安排工作，等他回来后，跟他汇报，发现是有差距的，并且有些工作还得他出面协调。为什么我做的不够好，为什么我现在做不了这个组的主管。一个星期的时间，我认识到，能力是一方面，上下文也是一方面，我缺少他知道到的信息，这也会影响决策的方方面面，从此，开始不断拓展自己的能力圈。

2019-01-09 12:26

作者回复

你已经具备了向前一步的基础，赞！

2019-01-09 18:32



喜悦

今日概念：

1. 工作的上下文：有点类似开发环境，所处的环境不同解决问题的方法就不一样；

今日总结

手里拿着锤子，看什么都是钉子。

这样做是因为自己掌握的工具（包括思维）不够。能摆脱自身环境的束缚，从更高的维度解决问题是一种向上能力。我一直相信这样的观点：

想要到达什么位置，就先像坐在个位置的人那样做事、思考。

2019-01-10 23:42



Alexdown

更新请加微信1182316662 众筹更多课程210



还没满足当前需求，就以可扩展性之名考虑后需求，纠结着想要一次性写出完美的代码，没错就是我本人了！

老师的那位同事真是幸运能遇到您！

我一直在纠结着，实现一个功能可能有ABC三种方式，当量级上来后C更容易扩展成D满足后续需求，AB较难或需推倒有两个纠结点：

可能只有AB，根本没C自己在那里瞎转空耗

意识到了有ABC，但纠结着无法做出选择，因为抱着那位同事的心态想一次搞定完美解决，可惜能力不够不知道C能扩展成D

希望听听老师的建议，谢谢

2019-01-10 19:33



香烟情人

郑老师这个观点让我醍醐灌顶，其实日常工作中我们已经不自觉的在受上下文这个底层规律的约束，上下文范围大的人视野会更广，格局会更大，考虑问题自然更周全，上下文小的人很可能只是守着自己的一亩三分地，甚至坐井观天而不自知。

2019-01-09 23:21



kissingers

就是要扩大视野，解决问题时跳出自己的职位来看看

2019-01-09 12:37



conanxu

想起来面试的时候，技术方案并没有怎么被挑战，但是这个功能到底有没有被用到被挑战了很多次。

2019-01-09 09:40