

03讲DoD的价值：你完成了工作，为什么他们还不满意



开始今天的讨论之前，我们先来看一个小故事。小李是一个程序员，有一天，项目经理老张来到他身边，和他商量一个功能特性的进度：

老张：这有一个任务需要完成，你看一下。

小李：这个不难，两天就能做完，两天以后就能上线。

两天以后，老张又来到小李的身边验收工作：

老张：怎么样，做完了吗？今天能上线吗？

小李：我的代码写完了。

老张：测试人员测过了吗？

小李：还没有。

老张：那今天能测完吗？

小李：那我就知道了。

老张：什么？我可是答应了业务的人，今天一定要上线的！

很明显，老张有些愤怒，而小李也有些委屈。于是，老张、小李和测试人员一起度过了一个不眠之夜。

听完这个故事，你有什么感想呢？先不急，我们继续看后面的故事。

又过了几天，老张又来找小李，给小李安排一个很简单的功能。在小李看来，一天就能搞定，而按照老张给出的时间表，小李有两天时间处理这个功能。小李心中暗喜：看来我可以“偷得浮生一日闲”了。

两天以后，老张又来检查工作。

老张：这个功能开发完了吗？

小李：写完了，你看我给你演示一下。

小李熟练地演示了这个新写好的功能，这次老张很满意：

老张：做得不错。单元测试都写了吧？

小李：啊？还要写单元测试吗？

老张：要不为啥给你两天的时间？

怎么会这样？小李心里很委屈，自己明明已经很好地完成了工作，老张是不是故意在找自己的麻烦呢？

好，故事讲完了。是不是有些似曾相识的感觉呢？为什么小李辛辛苦苦地工作，老张却总能挑出毛病来呢？老张是不是来挑刺的呢？其实，老张才没那么闲，小李的委屈主要是因为他和老张对于“完成”有着不一样的理解。换句话说，他们之间存在一个理解的鸿沟。

理解的鸿沟

在这个模块里，我们讨论的主题是“以终为始”。那我们第一个问题就是，“终”到底是什么？在前面这个例子里，“终”就是“完成”，可是，小李认为他的活已经做完了，老张却认为他没做完。

怎么会这样？二人之所以有分歧，归根结底，就在于二人对“完成”的定义理解的不同。

在第一个故事里，作为项目经理，老张认为“完成”应该是“上线运行”，而程序员小李则认为“完成”是“功能代码编写完毕”。这中间存在的理解偏差，包括了测试人员的测试工作，可能还包括了运维人员的上线工作。

在第二个故事里，老张给了小李两天时间。小李认为这两天都是编写功能代码的，而老张想的是，小李应该自己写好功能代码和单元测试，可能还包括了功能测试，这中间的差异是测试代码的工作量。

因为双方的理解不一致，所以无论怎样努力，小李都不可能达成项目经理老张的要求，正所谓“南辕北辙”。

那该怎么办呢？小李会说，我又不是老张肚子里的蛔虫，怎么才能和他达成一致呢？答案很简单，既然双方的理解有差异，那就把这个差异弥合上，后面的问题便也不是问题了。

弥合差异的方式有很多，有一个最佳实践，它的名字叫 DoD（Definition of Done，完成的定义），从这个概念的名字便不难看出，它就是为了解决软件开发中常见的“完成”问题而生的。

完成的定义

DoD 这个概念本身并不复杂，它就是告诉我们怎样算是完成了，尽量减少因为理解偏差造成的各种浪费。具体怎么做呢？就是团队在开始工作前，先制定 DoD。以前面的场景为例，团队可以规定：

特性开发完成，表示开发人员经过了需求澄清、功能设计、编写代码、单元测试，通过了测试人员的验收，确保代码处于一个可部署的状态，相关文档已经编写完毕。

开发完成，表示开发人员编写好功能代码，编写好单元测试代码，编写好集成测试代码，测试可以通过，代码通过了代码风格检查、测试覆盖率检查。

大家都是聪明人，一旦 DoD 确定好了，谁该做什么事就一目了然了。这个时候，如果小李说“我已经开发完了”，却只是写好了功能代码，那就别怪老张手下无情了。

好了，你已经知道 DoD 是什么了，它简单到让人一目了然，相信你很快就能知道怎样把它用到你的工作里。不过，我们不仅要知道怎么用，还要知道怎样让 DoD 更好地发挥作用。

- **DoD 是一个清单**，清单是由一个个的检查项组成的，用来检查我们的工作完成情况。DoD 的检查项，就是我们开发产品所需的一系列有价值的活动。比如：编写代码、编写测试代码、通过测试人员验收等。什么样的活动是有价值的，也许每个

团队的认识是不同的。但如果你的团队认为除了功能代码，其他都没价值，也许这是个信号，说明你的团队整体上是缺乏职业素养的，在这样的团队工作，前景并不乐观。

- **DoD 的检查项应该是实际可检查的。**你说代码写好了，代码在哪里；你说测试覆盖率达标了，怎么看到；你说你功能做好了，演示一下。
- **DoD 是团队成员间彼此汇报的一种机制。**别把“汇报”想复杂了，最简单的汇报就是说一句“这个功能做完了”。**当我们有了 DoD，做事只有两种状态，即“做完”和“没做完”。**在团队协作中，我们经常会听到有人说“这个事做完了80%”，对不起，那叫没做完，根本没有80%做完的说法。

在前面的讨论中，我们所说的 DoD 只是从个人层面入手。在团队层面，我们也可以定义 DoD。

- 某个功能的 DoD，比如：这个功能特性已经开发完成，经过产品负责人的验收，处于一个可部署的状态。
- 一个迭代的 DoD，比如：这个迭代规划的所有功能已经完成。
- 一次发布的 DoD，比如：整个软件处于可发布的状态，上线计划已经明确。

站在 DoD 的肩膀上

至此，我们只是从软件开发团队内部协作的角度来谈 DoD。但实际上，它不仅局限在团队内部协作上，如果你可以放开思路，会发现 **DoD 的思维在工作中用途非常广泛**。比如，当我们需要和其他团队合作开发一个接口时，我们都知道第一步就是要把接口定义下来。

那么，怎样才算定义完成？很多团队认为落在字面上就够了。但是有了 DoD 的思维，我们定义接口，就会去明确定义可检查的检查项。那么在定义接口这件事上，什么才是“可检查”的呢？我们可以参照一个可运行的接口来进行评估。只要检查：

- 服务方提供的接口是不是和这个可运行的接口返回值是一样的；
- 调用方是否可以和这个可运行的接口配合使用。

谁错了，谁改去。你可能会问，应该参照哪些可运行的接口呢？这不难解决，现在模拟服务器的框架到处都是。如果你不介意的话，我的 Moco 就是这样一个开源项目，你可以看一下。

在协作中一旦确立好 DoD，我们甚至可以通过流程把它固化下来，从而更高效高质地完成工作。当然，我们在工作生活中难免会有一些临时的工作，它们没有复杂到需要一个流程，但是也可以用 DoD 思维来高效地解决。比如：

经常会有人过来，让我帮忙做些事。运用 DoD 的思维，我首先会问他我具体要做哪些事，确认好细节（相当于定义好“检查项”），然后我就知道，这个忙我能帮到什么程度。

我请别人帮忙的时候，也会很清楚告诉他，哪些事是需要他做的，尽量减少不必要的误解。

DoD 是一个的思维模式，是一种尽可能消除不确定性，达成共识的方式。我们本着“以终为始”的方式做事情，DoD 让我们能够在一开始就把“终”清晰地定义出来。

人与人协作中，经常会出现各种问题，根本原因就是，有太多因为理解差异造成的误解，进而浪费了大量的时间，而 DoD 就是一种将容易产生歧义的理念落到实处的方法。

总结时刻

好，我们来总结一下今天学到的内容。首先，你应该知道，人与人协作，总会有这样或那样的理解差异。开始协作之前，我们最好先同步一下彼此的理解，确保之后不会因为理解不一致，而让协作方措手不及。

怎样解决大家的理解偏差呢，我介绍了 DoD（完成的定义），它是行业中的一种最佳实践，能够在团队内部很好地同步大家

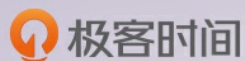
对“完成”的理解。好的 DoD 是一个可以检查的清单，可以确保你不遗漏任何事情。

如果深入领会 DoD，你会发现 DoD 可以灵活应用在不同的协作场景中。比如应用于个人工作、团队工作，甚至跨团队工作。当然，你也可以将它灵活地运用于各种生活场景，弥合人与人理解之间的差异，更好地协作与沟通。

如果今天的内容你只能记住一件事，那请记住：**在做任何事之前，先定义完成的标准。**

最后，我想请你回想一下，你在工作或生活中，是否发生过因为双方理解差异导致的问题或不快呢？有了 DoD 的概念以后，你是不是有了一些新的想法呢？欢迎在留言区留言。

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



雍

公司有各种checklist，类似于DoD，举例开发过程自检就有41项，开发人员真要check完，至少要花半天时间，大多时候开发都是复制以前的，然后“偷得浮生半日闲”了，项目组成员一多，全部复核一遍基本不可能，如何确定真正落实checklist就成个问题了，老师有没有什么好的建议？

2018-12-31 10:06

作者回复

我很好奇，四十多项都检查什么，是否可以分享一下？

2019-01-01 21:41



Alexdown

写的太好了，谢谢老师帮我捋清头脑中模糊不清的东西

前面那两个故事简直就是我和领导之间的情景再现，我的“完成”——实现功能，通过单元测试，他的“完成”——响应时间，系统吞吐，部署上线。

当你问他他的“完成”定义标准时，在得到解答后还会得到一个“这还用说的那么细吗，这不都应该知道的吗”排斥感，很反感这种感觉，每次问点啥后都有被鄙视被diss的感觉，不知该如何处理

2018-12-31 09:52

更新请加微信1182316662 众筹更多课程180



David Mao

职场新人估计都会遇到文中所描述的场景，上下级没有做好DoD的沟通，其实这是上下级都需要的职场素养。

1. 上级交代任务时把完成的指标和工作的边界讲清楚，下级根据要求去做。
2. 反过来，下级应不断提高技能，能自己定义DoD，请上级check，毕竟上级不希望每次沟通都占用较多时间。

2019-01-01 10:44



Demi

我们在工作中，都不会去写单元测试，认为那是测试人员的事。也没有时间去写，写功能代码就已经要加班加点了。可能也是因为懒，公司没有强制写，只要代码能跑通，没有很大的BUG，就可以上线。我不知道其他公司是不是这样。我做Android开发3年了，待过两家公司。第一家创业干了四个月，公司倒闭了。然后目前这家公司呆了两年多了。老师您这么一分析，我得提高下职业素养了。

2018-12-31 07:33



北天魔狼

偶像，醍醐灌顶啊。一开始就要确定完成的定义。

2018-12-31 06:40

作者回复

加油！

2019-01-03 18:56



喜悦

今日概念：

1. DoD (Definition of Done)：DoD是一种思维模式，是一种尽可能消除不确定性，达成共识的方式；
 - DoD 是一个清单，清单是由一个个检查项组成的，用来检查我们的工作完成情况；
 - DoD 的检查项应该是实际可检查的；
 - DoD 是团队成员间彼此汇报的一种机制；
2. 任务开发只有两种状态：完成/未完成；

今日总结：

1. 使用DoD可以事先将需要做的事量化；
2. 一件事用了DoD之后，就确定了需要完成的具体事项，不会觉得自己有很多时间，提高效率；
3. 使用不同“粒度”定义DoD有助于换位思考，也可以加深对系统的理解；

使用DoD的例子

1. 做计划，比如制定年度计划；

2019-01-02 22:52



Monday

正在立2019的flag。刚好可以使用DoD

2018-12-31 11:54



彩色的沙漠

Dod，做任何事情之前，先定义完成的标准。目前工作中有一项使用了Dod，提测标准：开发人员提测之前，要自测通过测试人员针对本次开发梳理出来的自测的case，测试人员也以这个自测case作为检查项，达不到提测标准的退回，开发人员重新测试。

作为已经工作了四年的开发人员来说，对于完成还是停留在功能代码写完，令人汗颜，说实在的自己也不会写单元测试和集成也没有去了解过，造成改完代码覆盖测试的成本极大，今年需要弥补单元测试和集成测试自己自动化测试的短板。

2019-01-02 20:52



pyhhou

老师想请教一下，现在我在一家小公司里工作，人很少，没有项目经理，直接听老板讲需求，老板不怎么懂技术，没有什么技术基础，对一个项目需要的时间以及技术流程不太清晰，加上我也是刚从学校里出来可以说没啥经验，他问我多久能完成我也是迷迷糊糊的给不出一个大概的时间，老板需求也会时常变化即便是经常开会说想法讨论方案，没有严格的DoD，可能就像您说的完成功能代码就算完成，因此我们的产品老是出问题，想请问下在这种环境下如何更好的使用DoD呢，或者说是更好的提升工作效率，和上级沟通呢？是不是只要是时间足够就尽量把自己的DoD定严一些，或者是根据时间来定DoD？谢谢老师

2018-12-31 22:31

作者回复

遇到一个不懂技术的老板是一件让人头疼的事。你可能需要拉着老板一起坐下来，先分析问题在哪里，再来谈如何改进。后面会讲到复盘与回顾，到时候，你可以尝试一下。

更新请加微信1182316662 众筹更多课程181

2019-01-01 21:36



雷小鸿

完成：先定义完成标准。

2019-01-07 13:54



飞扬

特性开发完成是指主要功能开发完成吗？

2019-01-06 08:42



Li

如果音频文件有进度条可以拖动更好

2019-01-04 16:13

作者回复

有的，点击右上角音频的符号就能进去。

2019-01-04 19:02



oddrock

本小节关键词:DoD

2019-01-04 08:16



一步

完成赞同 DoD的价值

2019-01-03 19:18



彩色的沙漠

DoD，做任何事情之前，先定义完成的标准。目前工作中有一项使用了DoD，提测标准:开发人员提测之前，要自测通过测试人员针对本次开发梳理出来的自测的case，测试人员也以这个自测case作为检查项，达不到提测标准的退回，开发人员重新测试。

作为已经工作了四年的开发人员来说，对于完成还是停留在功能代码写完，令人汗颜，说实在的自己也不会写单元测试和集成也没有去了解过，造成改完代码覆盖测试的成本极大，今年需要弥补单元测试和集成测试自己自动化测试的短板。

2019-01-02 20:41



小可

DoD，我觉得不仅仅用在程序开发方面，其实各个沟通做什么的场景都可以用到。比如，老板给小王说把考勤弄一下，小王就从考勤机里把考勤记录导出来发给老板，而老板要的是谁这月迟到了几次，请假了多久的统计表，结果可想而知。

另外我觉得，在交代或者沟通做什么不仅仅是接受方要明确和确认达到的目标，任务的提出方，也要尽可能表达清楚，避免沟通出差，不要嫌麻烦，比起看到不想要的结果，这点都不算事。

2019-01-02 08:45



achenbj

嗯，刚换公司，可以尝试。

2019-01-02 07:38



高阳路人

感觉有点像OKR，老师是不是也会讲？

公司统一定义的检查项有点像最佳实践，在落地时要根据实际情况去裁剪，生搬硬套要么流于形式，要么官僚主义。

2019-01-01 21:31

作者回复

根据自己的情况调整是一个好的做法，这是我要讲这些实践背后逻辑的原因。

2019-01-02 22:50



光明

【理解的差异】是造成开发资源浪费的主要原因之一。以前也听说过 对一个事情需要反复确认，就是尽量避免理解差异。现在有了DoD，真的可以在做任何事情前想好，什么是完成？做到什么程度？多方达成共识，协作就很方便了。

2019-01-01 11:50

作者回复

DoD作为一个实践，强调的是动作，而不是所有内容。如果是需求层面的东西，那是后面会讲到的内容。

2019-01-01 21:32



阿狸爱JAVA

一开始钢筋公司的那会，组长让我梳理一个下单的流程，我以为他说的是创建订单这个功能，而到最后才发现他说的是经过各个审批之后的下单。由此增加了我和他之间的埋怨。类似这样的事情发生过多次，在工作中，总是因为各种原因没有做DoD，又或者比如接口，明明做了参数和返回值的定义，但对方依旧会在开发过程中对其进行改动，导致我针对该接口的调用又要

重头再来。

2018-12-31 16:50

作者回复

作为一个专业人士，少不了要和不专业的人打交道，一个方法是尽早把问题暴露出来，让大家知道这是有问题的，然后再谋求改进。在沟通反馈部分，我会谈到相关话题。

2019-01-01 21:39