

08 | 为什么说做事之前要先进行推演？

郑晔 2019-01-11



00:00

讲述：郑晔 大小：11.41M 时长：12:27

1.0x

你好，我是郑晔。

经过前面的学习，想必你已经对“以终为始”这个原则有了自己的理解。你知道接到一个任务后，要做的不是立即埋头苦干，而是要学会思考，找出真正的目标。那目标明确之后，我们是不是就可以马上开始执行了呢？

先不着急给出你的答案，今天的内容从一个技术任务开始。

一个技术任务

你现在在一家发展还不错的公司工作。随着业务的不断发展，原来采用的关系型数据库越发无法满足快速的变化。于是，项目负责人派你去做个技术选型，把一部分业务迁移到更合适的存储方式上。

经过认真的调研和思考，你给负责人提出了自己的建议，“我们选择 MongoDB。”出于对你的信任，负责人无条件地同意了你的建议，你获得了很大的成就感。

在你的喜悦尚未消退时，负责人进一步对你委以重任，让你来出个替代计划。替代计划？你有些不相信自己的耳朵，嘴里嘟囔着：“把现在存到数据库的内容写到 MongoDB 不就成了，我就一个表一个表地替换。难道我还要把哪天替换哪个表列出来吗？”

刚刚还对你欣赏有加的负责人，脸色一下子沉了下来。“只有表改写吗？”他问你。你一脸懵地看着他，心里想，“不然呢？”

“上线计划呢？”负责人问。

“我还一行代码都没写呢？”你很无辜地看着负责人。

“我知道你没写代码，我们就假设代码已经写好了，看看上线是怎样一个过程。”

“不是发新版本就好了吗？”你还是不知道负责人到底想说什么。

“你能确定新版代码一定是对的吗？”

虽然你已经叱咤编程很多年，但作为老江湖，一听这话反而是有些怯的。“不能。”你痛快地承认了。

“一旦出错，我们就回滚到上一个版本不就成了。”常规的处理手段你还是有的。

“但数据已经写到了不同的存储里面，查询会受到影响，对不对？”负责人一针见血。

“如果这个阶段采用两个数据存储双写的方案，新代码即便出问题，旧存储的代码是正常，

我们还有机会回滚。”你一下子就给出了一个解决方案，咱最不怕出问题了。

“对。”负责人认同了你的做法，一副没看错人的神情。“让你出上线方案，就是为了多想细节。”

你终于明白了负责人的良苦用心，也就不再大意。很快，你就给出了一份更详尽的上线方案。

事项	备注
上线双写版本	如果有问题，立即回滚
双写版本稳定运行一段时间	确认写新存储的方案一切正常
上线从新库读取的方案	如果有问题，立即回滚
读取版本稳定运行一段时间	确认读新存储的方案一切正常
下掉写老库的版本	如果有问题，立即回滚

你把这个方案拿给负责人看，信心满满，觉得自己够小心，一步一步做，没有任何问题。但负责人看了看你的上线计划，眉头逐渐锁了起来，你知道负责人还是不满意，但不知道还差在哪里？

“原有的数据怎么办？”负责人又问了一个问题。你一下子意识到，确实是问题。“没有原有数据，一旦查询涉及到原有数据，查询的结果一定是错的。所以，还应该有一个原有数据的迁移任务。”你尴尬地笑了笑。

负责人微笑着看着你。“好吧，从我的角度看差不多了，你可以再仔细想想。然后，排一个开发任务出来吧！”

你当然不会辜负负责人的信任，很快排出了开发任务。

任务
添加一个读写新库的组件
根据配置确定是否启动新库的读写
根据配置确定是否启动老库的读写

看着排出的任务，你忽然困惑了。最开始只是想写个读写新库的组件，怎么就多出这么些任务。此外，你还很纳闷为什么负责人总是能找到这么多问题。

一次个人回顾

你想起之前的工作里有过类似的场景，那个负责人也是让你独立安排任务。通常，你最初得到的也是一个简单的答案，从当时的心境上看，你是很有成就感的。

只是后来的故事就不那么美妙了，上线时常常出现各种问题，你和其他同事们手忙脚乱地处理各种异常。当时顶着巨大压力解决问题的场景，你依然记忆犹新。解决完问题离开公司时，天空已经泛起鱼肚白。

而似乎自从加入了现在的公司，这种手忙脚乱的场景少了很多。你开始仔细回想现在这个负责人在工作中的种种。从给大家机会的角度来看，这个负责人确实不错，他总会让一个人独立承担一项任务。只不过，他会要求大家先将任务分解的结果给他看。

拿到组里任何一个人的开发列表之后，他都会问一大堆问题，而且大多数情况下，他都会问到让人哑口无言。说句心里话，每次被他追问心里是挺不舒服的，就像今天这样。

本来在你看来挺简单的一件事，经过他的一系列追问，变成了一个长长的工作列表，要做的事一下子就变多了。毕竟谁不愿意少做点活呢！

不过，你不得不承认的一点是，加入这个公司后，做事更从容了。你知道无论做的事是什么，那些基本的部分是一样的，差别体现在事前忙，还是事后忙，而现在这家公司属于事前忙。于是，你开始把前一家公司上线时所忙碌的内容，和现在负责人每次问的问题放在一起做对比。

这样一梳理，你才发现，原来负责人问的问题，其实都是与上线相关的问题。包括这次的问题也是，上线出问题怎么办，线上数据怎么处理等等。

你突然意识到一个关键问题，其实负责人每次问的问题都是类似的，无论是你还是其他人，他都会关心上线过程是什么样，给出一个上线计划。即便我们还一行代码都没有，他依然会让我们假设如果一切就绪，应该怎样一步一步地做。

你终于明白了，之前的项目之所以手忙脚乱，因为那时候只想了功能实现，却从来没考虑过上线，而且问题基本上都是出在上线过程中的。你想到了上次参加一个社区活动，其中的一个大牛提到了一个说法：“**最后一公里**”。

想到这，你赶紧上网搜了一下“最后一公里”，这个说法指的是完成一件事，在最后也是最关键的步骤。你才意识到，“最后一公里”这个说法已经被应用在很多领域了，负责人就是站在“最后一公里”的角度来看要发生的事情。

嗯，你学会了一招，以后你也可以站在“最后一公里”去发现问题了，加上你已经具备的推演能力，给出一个更令人满意的任务列表似乎更容易一些。

把这个问题想清楚了，你重新整理了自己的思路，列出了一个自己的问题解决计划。

- 先从结果的角度入手，看看最终上线要考虑哪些因素。
- 推演出一个可以一步一步执行的上线方案，用前面考虑到的因素作为衡量指标。
- 根据推演出来的上线方案，总结要做的任务。

不过，更令你兴奋的是，你拥有了一个看问题的新角度，让自己可以再上一个台阶，向着资深软件工程师的级别又迈进了一步。

通往结果之路

好了，这个小故事告一段落。作为我们专栏的用户，你可能已经知道了这个故事要表达的内容依旧是“以终为始”。关于“以终为始”，我们前面讲的内容一直是看到结果，结果是重要的。然而，**通向结果的路径才是更重要的。**

这个世界不乏有理想的人，大多数人都能看到一個宏大的未来，但这个世界上，真正取得与这些理想相配成绩的人却少之又少，大部分人都是泯然众生的。

宏大理想是一个目标，而走向目标是需要一步一个脚印地向前走的。唐僧的目标是求取真经，但他依然用了十几年时间才来到大雷音寺。唐僧西天取经有一个极大的优势，他达成目标的路径是清晰的，从长安出发，向着西天一路前行就好。

对比我们的工作，多数情况下，即便目标清晰，路径却是模糊的。所以，不同的人有不同的处理方式。有些人是走到哪算哪，然后再看；有些人则是先推演一下路径，看看能走到什么程度。

在我们做软件的过程中，这两种路径所带来的差异，已经在前面的小故事里体现出来了。一种是前期其乐融融，后期手忙脚乱；一种是前面思前想后，后面四平八稳。我个人是推崇后一种做法的。

或许你已经发现了，这就是我们在“以终为始”主题的开篇中，提到的第一次创造或者智力上的创造。如果不记得了，不妨回顾一下 [《02 | 以终为始：如何让你的努力不白费？》](#)。

实际上，早就有人在熟练运用这种思想了。**在军事上，人们将其称为沙盘推演，或沙盘模拟。**军队通过沙盘模拟军事双方的对战过程，发现战略战术上存在的问题。这一思想也被商界借鉴过来，用来培训各级管理者。

这个思想并不难理解，我们可以很容易地将它运用在工作中的很多方面。比如：

- 在做一个产品之前，先来推演一下这个产品如何推广，通过什么途径推广给什么样的人；
- 在做技术改进之前，先来考虑一下上线是怎样一个过程，为可能出现的问题准备预案；
- 在设计一个产品特性之前，先来考虑数据由谁提供，完整的流程是什么样的。

最后这个例子也是软件开发中常遇到的，为数不少的产品经理在设计产品时，只考虑到用户界面是怎样交互的，全然不理睬数据从何而来，造成的结果是：累死累活做出来的东西，完全跑不通，因为没有数据源。

很多时候，我们欠缺的只是在开始动手之前做一遍推演，所以，我们常常要靠自己的小聪明忙不迭地应对可能发生的一切。

希望通过今天的分享，能让你打破手忙脚乱的工作循环，让自己的工作变得更加从容。

总结时刻

即便已经确定了自己的工作目标，我们依然要在具体动手之前，把实施步骤推演一番，完成一次头脑中的创造，也就是第一次创造或智力上的创造。这种思想在军事上称之为沙盘推演，在很多领域都有广泛地应用。

在软件开发过程中，我们就假设软件已经就绪，看就绪之后，要做哪些事情，比如，如何上线、如何推广等等，这样的推演过程会帮我们发现前期准备的不足之处，进一步丰富我们的工作计划。为了不让我们总在“最后一公里”摔跟头，前期的推演是不可或缺的，也是想让团队进入有条不紊状态的前提。

如果今天的内容你只记住一件事，那请记住：**在动手做一件事之前，先推演一番。**

最后，我想请你思考一下，如果把你在做的事情推演一番，你会发现哪些可以改进的地方呢？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。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

该试读文章来自付费专栏《10x程序员工作法》，如需阅读全部文章，
请订阅文章所属专栏

立即订阅



聚丙烯🌸 收割机配件

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。

Ctrl + Enter 发表

0/2000字

提交留言

精选留言(36)



ZackZzzzz

项目上线之前，一般都会有一个launch plan, 数据库迁移这种项目，不去考虑上线回滚我认为是设计上的缺失。我们公司的launch plan一般是写成一步一步的checklist, 在上线之前会做同伴审查

作者回复: 做得很棒，值得分享!

2019-01-11



👍 25



adang

想清楚了才能写清楚，这是我在编程工作非常认可的一句话，并且我也认为它是区分合格与不合格开发工程师的重要区别。软件开发过程中，最常见的例子就是拿到需求后不管三七二十一，上来就开始撸代码，但最后往往返工不断，质量问题层出不穷，而且加班没完没了，这里面一个根本原因就是没有系统地想清楚，但很多人都觉得前期澄清需求、分析设计是浪费时间，只有编码才是真正的创造价值，这就是差距。

谢谢老师今天的分享，深有同感。

2019-01-11



👍 17



休息一下马上回来

上线前，哪些机器什么配置应该有一个预期甚至提前准备好

作者回复: 很好的补充

2019-01-11



8



Scott

我觉得领导说先跑通再说和事前推演是不矛盾的，很多时候，我们需要一个poc来证明这个项目是可行的，这其实也是事前推演的一部分。

上线要事无巨细的检查推演，和快速跑通poc不矛盾，当然现实世界是，大家就急着把poc当正式产品上线了，这是无数个悲剧故事的序章。

作者回复: 对，能分清楚PoC和产品是两个不同东西的团队，都是好团队。PoC就是该丢弃的，包括代码，因为它的代码质量是不合格的。

2019-01-12



7



Alexdown

非常喜欢老师这种对话式职场故事风格！

曾经对领导问东问西，他说：“先把眼前事做好，先跑通再说……”，听完后我就吭哧吭哧地开始实现，想要跑通后做出来看看，然后各种不对。

现在想想我是会错意了，领导说的是不要过度设计，只要满足当前需求就好，不要在脑中脑补未来可能出现的高并发大流量的需求。而事前分析即事前推演该做还得做，为了“满足当前需求”而做！

作者回复: 分析和设计是少不了的，不发散也是重要的。

2019-01-11



5



Kăfkă²⁰²⁰

推演可以发现达成目标会涉及到哪些部门、哪些利益相关者，需要哪些资源，以及他们需要何时怎样的配合。

2019-01-11



3



sam

最近负责一个项目对接，计划把对方一个内网应用服务接入到移动端应用平台上，排期2~3天，结果联调过程中，新的问题不断发现，从原有的内网应用，到经过防火墙、nginx代理的外网请求，内外网域名变了，安全策略也变了，网络环境变了，半个多月过去了，还在延期。
这个过程，如果要做沙盘推演，是需要对两边整个技术架构都了解的人，也才能事前规划好。

作者回复: 不清楚的事，需要有一个计划，把事情弄清楚。

2020-06-30



1



helloworld

好文！在开始之前先做推演的工作方式非常好，可以防止工作疏漏更重要的是可以提前知道预先设定的方案到底不可行

作者回复: 把事情做在前面，减少后续的忙碌。

2019-02-20



1



大彬

我们工作、生活都是有目标的，达到目标有2种方法：A. 先规划好到目标的路线，一步一步往下做。B. 没有规划完整路线，而是依据当前的情况去做。A方案也许会遇到未曾预料的问题，但总体比较顺畅，能达成目标。B方案一路披荆斩棘，累死累活，结果可能还不能及时达成目标。想想工作中没做好的任务，是不是前期没有好好准备？

作者回复: 一个正常的做法应该是，有方法按照方法做，没办法再想办法。可是，很多人把所有

事情都做成了现想办法，那就会很辛苦。这种做法就像低等生物一样，完全靠外界刺激。

2019-01-15



👍 1



喜悦

今日概念：

1. 最后一公里：最后一公里就是离“终”最近的那一部分；
2. 个人回顾：也就是复盘，复盘是积累和反思；

今日总结：

重视最后一公里可以跟细节的考虑我们要得“终”，面向“终”分析业务过程，会更清楚的知道要做什么；

2019-01-13



👍 1



冰糕不冰

最近一次上线，开发测试运行良好！生产环境上去就发现速度很慢，由于表数据太多了。事前推演，站在最后一公里考虑问题。宁愿事前辛苦，也不要事后遭罪。

2019-01-13



👍 1



2019

磨刀不误砍柴工！

2019-01-13



👍 1



null

可是小公司经常遇到的问题是，老板随时都在变需求，每次需求又很急，我们给的方案建议他不采纳，说我们不了解行业，但是他又经常变

作者回复: 别急，答疑我详细回答了和上级的关系，如果还有问题，我们再来讨论。

2019-01-12



👍 1



公号-技术夜未眠

以终为始，通往目标的要事第一。

作者回复: 什么是要事，我们在 任务分解 模块聊。

2019-01-12



👍 1



號國技醬

打卡

2019-01-11



👍 1



oddrock

主题：做事之前的推演

关键词：第一次创造，最后一公里

2019-01-11



👍 1



2021年薪资目标35K

凡事预则立 不预则废

2020-11-21



👍



2021年薪资目标35K

上家公司专门有个需求“发布计划”服务，产品、开发、测试每个角色的任务需求在发布前装备好并保存到发布系统上。创建好会自动拉群，每个人的任务状态变化事件都会实时的推送到发布群。

2020-11-21



👍



请叫我和尚

系统的容灾性，其实也属于推演中的一个 checklist 吧

2020-11-20



yudidi

mark一下。

老师本文所说的推演，其实比我们平常的推演会更细致一点，是落实到执行层面的。不仅仅是一个简单的预演。

作者回复: 没问题，越细致越好。

2020-09-10



yudidi

在一个中小公司工作，有些部门就有上线后手忙脚乱的情况。我们是一个新成立的部门，做事情的时候相对比较重了，现在想来应该就是做事之前先推演的思路。

不过我们之前并没有明确的推演意识。

看了老师的文章之后，刻意地去运用推演的方法，并且是更系统的推演方法，感觉收获很多，尤其是获得了这种方法论上面的收获。

非常感谢老师!!!

2020-09-10



sealeaf

前文小故事这简直就是我的原型啊

作者回复: 欢迎分享你的故事。

2020-07-07





escray

之前也经历过一些系统的升级改造，在上线的时候的确比较紧张，虽然也有类似 Launch Plan 和 Check List，但很多时候业务系统是没有办法回退的，或者说回退的代价太大。

作为旁观者，其实“主刀”在升级之前，也进行了过“推演”，开了各种协调会，但是事到临头还是会有各种意外。

Plans are nothing; planning is everything — Dwight D. Eisenhower.

但是推演似乎和这里的计划 Plan 又有所不同，老师似乎更强调“最后一公里”。

我觉的文中最有价值的部分在于：

做产品之前，先推演如何推广；
技术改进前，先推演上线过程；
设计产品特性，先推演数据来源；

如果有机会多次重复推演的话，那么应该就可以形成预案了。

作者回复: 计划一般是粗粒度的，而这里推演粒度要小得多，因为这是在执行层面。

2020-06-01



丁乐洪

checklist,不过我们都是上线前才准备，下次要提前

作者回复: 早做早发现问题

2020-05-23



wuhulala

其实拿到任务去做的时候，脑袋是不清晰的，可能会做一步，卡一步。现在通过思维导图以结果为中心，分解行动路径，等到真正做的时候，整个思路就会清晰很多，在遇到问题是，再把问题反馈到思维导图上面，周而复始，任务也晚了，思维导图也作为设计方案留下来了。

作者回复: 有些事早晚都得做，做得早了早发现问题，做得晚了慢慢填坑。

2020-03-19



吴凌华

问题是大多数时候，真正的产品经理是无知的老板

作者回复: 不捣乱的老板都是好老板。

2019-12-15



丁丁历险记

哈哈，我就是那位负责人。

2019-10-31



陈斯佳

以终为始不是目的，而是方法。真正的目的用以终为始的方法搭建走向目标的一步步路径。

2019-07-16



春之绿野

推演，这可能是我年度学到的最重要的一个概念吧，虽然以前也模模糊糊听说过这个词，但总觉得跟自己没啥关系，开会的时候大家讨论的很热烈，而我总是提不出什么有建设性的问题和建议，这个一直很困扰我，现在来看就是这种推演的能力不足，做事情只会走到哪算到哪。昨天开会前尝试了下推演，真的有很多发现，比如环境是瓶颈，这边的环境我不太了解，好像是因为要连很多硬件设备所以很有限，如果只有一套设备的话，自动化是否值得做。而且推演后也会对问题认识的更清晰更有把握了。以后一

定多练习推演，第一次创造。

2019-05-18



YiCo

任务分解，提前推演

2019-03-13



YiCo

任务分解，提前推演

2019-03-13



苦行僧

都是打实话为什么之前自己体会不到 这大概就是思维盲区的困惑 老师的文章 点醒梦中人

作者回复: 有个说法叫盲维，缺少了一个维度，即便简单，你依然看不到。

2019-03-07



苦行僧

现在很大城市都有公共自行车接近最后一公里问题，当然我们要避免的是提前预演自己的最后一公里能借到车

2019-03-07



111

推演也是在验证目标的可行性，也可以把可能遇到的问题提前考虑到，我们在建立表时经常用业务场景进行推演表的结构是否满足和可行性

作者回复: 拿结果看, 就知道自己做得如何了。

2019-02-24



莫小鹏

之前做的一个运营活动, 就出现手忙脚乱的场景。早点听到老师这个专栏就好了。

2019-01-12



墨鱼

前期的确定完成的具体目标很重要, 对于到前端我觉得应该把每一个story遇到的问题都推演一下。先选好方向。

2019-01-12

