

《电子书·软件测试综合能力提升》.pdf

你好，我是 IDO 老徐，感谢信任、感谢认可、感谢订阅。

电子书·软件测试综合能力提升：写了六年文章，书一直搁置，没出版，那就从电子书开始，定价 99 / 人

期望 1 万人订阅，如果我过往文章对你有用，欢迎随手订阅 1 份（365 篇更新完成后，会生成 1 份 PDF 版本，会签名，及写上你的专属编号 + 昵称，全网唯一）。

内容板块：

- 1、过往 10 年，IDO 老徐写的软件测试精选文章，精选 100 篇
 - 2、接触的十几万读者中，精选 100 个案例
 - 3、收到的几万个提问中，精选 165 个问题
- 如上，组成 365 篇文章，要么不发，发的就是精选

单向订阅，读者不支持发贴，但可互动（点赞，评论，提问）。可以把这里理解为一本可与作者 IDO 老徐 互动 的电子书

有疑问，微信 957863300 找 IDO 老徐（订阅后，请加 IDO 老徐微信。更新完结后，会送你一份 PDF 完整签名版，永久保存）

注：这是一份公开版，免费送给所有感兴趣的读者，是 520 的福利
(内容还没更新完，目前更新了 25 问 + 15 篇精选文)，总共 59 页

一、电子书·测试 165 问

001 问：冒烟测试，怎么测？

IDO 老徐，答：
概念看书，

如果实在不理解，从写好的测试用例里，找出 p0 和 p1 优先级的，测试通过，代表冒烟通过。

如果优先级也不理解，那就把主要功能流程跑一遍，能通 。

如果还不理解，找个同事带你吧。或者，转行 。

#测试 165 问#

002 问：现在各行各业内卷都比较厉害，不知道您对测试内卷有什么看法？希望能得到你的解答，谢谢。

IDO 老徐，答：
卷这个事，随她去吧 。

我对技术人的建议是：一家公司，除了拿薪资，
剩下的就是积累经验、提升能力，
干的不开心了，就换 。

#测试 165 问#

003 问：你好老徐，看你工作了那么久。我想问下，软件测试中有没有职场潜规则？有的话一般来说会遇到哪些，我们如何来闭坑，谢谢！

IDO 老徐，答：
1、Bug 记得走 缺陷系统
2、测试报告，注明风险，及已知问题

做好如上，避免背锅 。

至于其他的，
技术类岗位，比较单纯，好好做事，好好提升能力，两年左右一跳 。

#测试 165 问#

004 问：你好，最近特别迷茫，自己学了自动化和性能测试，面试各个公司都说要做，但是进入公司后基本不怎么做，学不能致用，导致学了一段时间之后就慢慢的遗忘掉了，感觉特别的无奈，然后又回归到了最初的起点，请问这种应该怎么办啊。

IDO 老徐，答：
尽量去 大公司、大团队、业务成熟点的 ，

外包、小作坊、创业公司，从效率和成本来看，能把业务测清楚就不错了，没时间让你折腾自动化，性能也没太大的需求（或有老大 兼着 搞定）。

建议，

- 1、虽然工作任务没有，但你自己，可以用所测的业务练手 接口测试、接口自动化。
- 2、尽量往大公司
- 3、合格业务测试工程师，参考文章 <http://istester.com/zhiye/365.html>

#测试 165 问#

005 问：日常工作中怎么去刻意锻炼测试工程师需要的各项能力？

IDO 老徐，答：

- 1、主动去思考
- 2、把遇到的每个问题，搞定（而不是因为不懂，退缩）
- 3、深究每个 Bug 的具体原因，并定位出来。
- 4、保持好奇心，去看服务端和客户端的 Bug，看看为什么有问题。

太多太多了。

这个问题，可以写几个专题文章了。

过往我写的参考文章：

- 1、知识体系：<http://istester.com/tester/304.html>
- 2、优秀的业务测试工程师：<http://istester.com/zhiye/365.html>
- 3、管理：<http://istester.com/manager/713.html>

#测试 165 问#

006 问：你好，我想问下现在测试招聘的时候对于培训班出身的非科班求职者的态度是什么样的？

像这类求职者面试的时候可以有什么技巧么？

IDO 老徐，答：

是个典型问题，之前文章也写过，

补充几个点：

- 1、我不太建议疯狂捏造简历经验（很多培训机构都是这样玩的，我不知道你的简历是否有这样的水分）
- 2、对于稍微有点经验的面试官，随便问几个问题，就知道水分了，会很尴尬。
- 3、但，确实会存在，没经验、没项目，很难有面试机会。

我的建议:

- 1、工作经验实事求是，项目经验，可以列几个实操的练习项目
- 2、把基础知识搞清楚，测试工具、sql、Linux、测试理论、http、语言基础 等
- 3、把自己简历上有的所有内容，搞清楚 。

#测试 165 问#

007 问：我目前工作 3 年，在岗位上目前属于资深测试工程师。其实自己想往管理上发展，那这样的话除了认真搞技术，那么管理方面需要去学习吗？请您给一个经验，谢谢。

IDO 老徐，答：

工作 3 年后，应该要去学习一些管理相关的知识了 。

一个好的学习思路：

多看看你的领导是怎么管理、怎么带这个团队的，以及你觉得他哪些点做的不好 。

看再多管理书，不如从身边学 。

注：具体管理建议，我之前写过很多文章，可以去搜索下

#测试 165 问#

008 问：请教一个职场问题。我今年 40 了，最近找到了一个「测试讲师」的工作，但是我比较纠结要不要换，毕竟这个年纪求稳。我现在在外包到一家银行，做了一年半了。讲师的工资比现在在多 5000，坐标成都，但是工作强度比较大，对我来说也是第一次做讲师，比较有挑战性 。

答：

- 1、如果 40 岁了，还在外包，相比讲师也许比外包更有意思，且薪资多 5000，
- 2、虽然都不咋样，但后者也许可尝试，毕竟没尝试过
- 3、严格来说，「讲师」这个岗位，会把人弄废，且转不了多少钱，我更建议 30 岁以后，玩自己的副业，培养个人 IP 。这是未来的大趋势 。

#测试 165 问#

009 问：老徐，测试工程师月薪 2 万 在深圳什么水平？

IDO 老徐，答：

1、稍微有点规模的企业，如果你是独立负责一个项目的测试（项目质量），且工作两年以上，都有可能达到 18K - 25K（这条，不仅在深圳，同样适用于北上广深）

2、但，从历年的数据来看，薪资 2 万以上已经很不错了，不低。

3、不要太关注是 18K，还是 20K、22K，要关注能力是否有提升，换一家公司，是否可以到 30K？

end

#测试 165 问#

010 问：研发提测质量低，每月统计质量数据后，研发没有任何的反馈；该做的都做了，怎么办？

IDO 老徐，答：

- 1、严把控「准入标准」
- 2、作为质量部门，每周 / 每月，反馈问题，出报告
- 3、至于解决这个问题；这是研发 Leader，或者技术部门 Leader，需要思考的事。
- 4、如果 研发 Leader、技术部门 Leader，没改进，可以考虑继续「向上反馈」。
- 5、如果你能做到「每个月，汇总统计，质量相应数据；如提测质量、提交 bug 数、修改缺陷效率、缺陷等级、各研发 Bug 数排名、Bug 类型分布 等」已经很不错了。如果能整理改进建议，更好。
- 6、如果这些都做到了，效果依然不理想（如，研发领导并不重视，质量数据依然很差），这种是公司的問題。
- 7、质量部门，能做的是「质量管理」（问题反馈，提建议），不是「提高研发代码质量」
- 8、很多时候，不是方法论问题，是「研发不改、公司不重视」

end

#测试 165 问#

011 问：对于一些难重现的 bug 该如何处理？

IDO 老徐，答：

1. 尽可能用各种技术手段定位（抓包，日志，埋点 等）
2. bug 系统，建专项跟进流程（也可以建专项 状态，具体看管理流程）

3. 持续几个周期，未复现，可关闭

012 问：因为一些原因，中间有几年没从事测试，现在还能找到工作吗？需要怎么做？

IDO 老徐，答：

这问题，非常典型，有创业几年失败回来的，有做小生意失败回来的，有全职带娃几年回来的，有转岗其他一圈后发现不适合回来的。

我的建议，

1. 没太大竞争力，但可尝试
2. 不去尝试，怎么知道能不能找到工作呢
3. 复习过往知识，投简历，面试
4. 其实，几年前启动的百人计划，就是解决这类综合能力提升的，具体可了解下

013 问：老徐，应届毕业生在这个公司快两年，是个外协职位，目前公司确实给了转正式的名额，但是转正的薪资没有其他同事跳槽的多，并且一直是功能测试，一点点 web 测试，接触自动化的机会非常少；有在考虑跳槽，你觉得是否值得留下？

IDO 老徐，答：

原则：先看看外面的机会，有适合的岗位且薪资满意，且拿到了 Offer，再考虑；

另，跳槽涨薪更快，很正常的现象，这是行业内都知道的事；

结论：骑驴找马，别裸辞。

014 问：老徐你好，2018 年毕业，在这家公司 2 年了，一直都在接触功能测试(APP)，平时在公司有研究接口自动化，已经有一套框架可以适用了(Python)，可是公司并不需要自动化，只需要点点点，平时都是在网上跟别人交流，很少跟公司的测试交流(加上我只有 3 个人)，一个人真的累了，也会用 Python 写一些测试工具，哄他俩用。这样的情况下要不要考虑跳槽了，去更好的平台？

IDO 老徐，答：

三人的团队，确实不需要太多的自动化测试 介入（核心接口自动化玩一玩，可以）；

这种场景下，业务测试价值更大、来的更快；

针对这个案例，@IDO 老徐 的建议：

1、日常测试工作中，有时间，可通过接口自动化，把自己负责的业务玩起来（公司是否需要不影响你自己搞，前提是业务测试做完的情况下，别本末倒置）； 即可以确保线上重点故障少，也可以让自己的工作更轻松点； 还可以提升自己的接口实战经验，多好；

想开了，就开心了；

2、至于是否要跳槽，还是昨天解答某同学的提问，同样观点：骑驴找马
<https://t.zsxq.com/RvNJmYJ>

3、最后，
想开了，每天多学点，找机会跳槽；

4、在当前公司没成长（除非公司接下来一年有规模成倍增长的可能性）；

015 问：新入职一家公司，只有我一个测试，什么流程都没有，乱，完全不知道如何怎样开始工作，怎么办？

IDO 老徐，答：

1、做好心理准备

这类公司特色，

- 1) 测试不受重视，可有可无。
- 2) 测试只是辅助，无任何流程，准备发版了，发个包给你，让你测，连需求都不知道。
- 3) 任何事情，好像都跟测试无关，开发也无视测试的存在，提的 Bug 也不改。感觉被孤立。

2、我给 4 条建议，

- 1) 与老大沟通，了解下招你进来的定位是什么？解决什么问题？
- 2) 梳理目前你看到的研发团队问题，或者质量问题，给出你的建议。
- 3) 梳理一套可行的流程，提高效率和质量（能落地就行，不一定要高大上）。
- 4) 多学习，主动融入&深入到项目中，了解需求，了解系统架构。

016 问：测试工程师是否有必要考取一些证书，增加面试含金量？

IDO 老徐观点:

- 1、从在职的角度，没啥用，除了某些公司证书会有补贴。
- 2、对于学生阶段，如果收费不是特别贵，考没坏处。
 - 1) 在学校有时间
 - 2) 考证过程，可巩固理论知识。

3、嗯，老徐，啥证书没考。
这些年，一样，混的好好的。

互动:

你考了哪些证书，有哪些实际用途，是否有这方面资料和经验总结？

常见的证书，比如：软件评测师、ISTQB、PMP 等

017 问：请问有软件测试的私活（兼职）吗？

IDO 老徐，答：

- 1、测试接私活这件事，不靠谱，放弃，忘了她。
- 2、少有的几个云测平台，体验下就行，赚钱，难。
- 3、自己线下接项目，也有，但频率比较低。
- 4、我的建议是，换一个思路，利用互联网，写作，搞副业（具体可以参考我的路径）。

018 问：开发提测的版本，测试这边不清楚是否有改动其他的，且未告知我想用版本与版本之间的对比工具或者类似的工具，这样的话可能测试会更全面，但是随之而来查看代码能力对于测试来说要求也会变高。
疑问：这种方式是否可行？有推荐类似的工具吗？望解惑，谢谢

IDO 老徐，答：

不太建议；

有门槛，且费时间。就算你对比出来了，如果是业务测试工程师，case 也很难完整覆盖。

更建议看 git 代码提交记录 及 流程上提测邮件里注明测试重点。

这样就差不多了，且更能落地

另，提测邮件里，肯定是有这次提测涉及的仓库的。实在想看，去看看这几个仓库，提交的版本对比。

关于漏测：新业务的完整 case + 老业务的回归 case（这是必须要做的）。

019 问：面试过程中，通过怎样的观察 及 问题，可以更好的「了解测试直属领导性格、能力边界（负责内容，日常工作）」？

IDO 老徐，答：

这是一个好问题，是 <https://t.zsxq.com/02RVbQNBQ> 精选文 的延伸；
这问题，比较难，IDO 老徐 思考了两天，迟迟没有回答。

给大家几个可行性的思路（我面试的无数案例 及 经验总结）

1、看感觉（面试交流的半小时，问问自己对这位 Leader 的第一感觉怎样？如果不好，大概率后续配合会出问题，需谨慎）。是的，就是这么随意，第一感觉太重要了。

2、问一下团队现在用到的工具，除业务测试、主要涉及的工作内容（了解下这个团队，是业务为主，还是 测开 + 业务）。

3、通过观察，看看这个 Leader 是随和型 的还是 一本正经 型（即：面试全程，微笑过否？；不同类型的 Leader 管理风格完全不一样）

4、问一下研发团队规模、公司主要的业务、后续半年的规划（看看这 Leader 是闷头做事，还是会关注公司发展）

注：这问题，想清楚再问，也许会不解答。

如上，差不多了。

020 问：请教下，收到的人职 Offer 没写薪资这是正常的吗？

IDO 老徐，答：

不正常，建议让重新发 Offer，没有薪资的 Offer 会有一堆麻烦事，且不合理。

一般来说，Offer 上 会有：入职部门、直属 Leader、入职岗位、岗位薪资。

如果遇到这种情况，可以联系 HR，让重新发一下入职 Offer。

021 问：先做接口测试还是功能测试？

IDO 老徐，答：

接口先行，冒烟通过，测功能。

但，实际情况：很多时候，是来不及做接口测试的，或不做接口测试，或上线后再补。其实也可以。

如果公司没有硬性的流程要求，怎么舒服怎么来，我们关注的是结果：测试效率高，线上漏测率低。

022 问：测试执行阶段，每天发送测试进展的时候都包含哪些内容呢？进度具体以什么形式展现呢？用进度百分比吗？

IDO 老徐，答：

- 1、发测试日报，其实比较费时间。
- 2、这个形式，更多是为了了解：你的测试进度是否有风险，开发的 Bug 修复是否阻塞测试，Bug 是否过多。
- 3、了解了目的，具体怎么做就很清晰了：1) 测试进度 x%，今天发现了 x 个 bug，验证了 x 个 bug，还有 x 个未修复的 bug，进度正常，无风险。
- 4、具体形式，不重要，团队达成一致，你领导满意的方式就行（可直接如上文字群里发，也可以弄一个固定模版发，也可以系统自动推送，也可以用邮件（形式无所谓））。

023 问：老徐，你觉得过了 35 是走技术方向还是走管理方向呢？

IDO 老徐，答：

- 1、如果你还在问某个技术是否适合入门、是否上手容易，建议放弃技术，你不适合，可以考虑管理，或转岗（参考精选文 <https://t.zsxq.com/02JiYViaQ>）。
- 2、完全不懂技术的管理，也挺难的。管理为主，但要了解每个技术的「优劣」及「能做什么事」，便于选型。
- 3、作为管理，技术不一定要部门最强的，但你能带的动他们：让技术牛人有成就感、用技术解决问题。
- 4、管理者不应该成为技术天花板，@IDO 老徐 的测试部门，技术比老徐强的多了去了。
- 5、从整个职业发展来看：如果你做不到国内头部公司的技术专家，技术再强，薪资也就那样。管理具备可行性（虽然很卷）。
- 6、技术，只是用来解决问题的。管理者，更重要的是整个测试领域的知识体系 + 决策能力 + 用人。

024 问：还在大学，看到某培训机构，1w 的课程要不要学？

IDO 老徐，答：

直接拒绝，不用考虑、不用纠结 。

- 1、大学四年是提升综合能力 和自学能力，而不是培养技工 。
- 2、真想培训，大四毕业后，也不迟（当然，严格来说，我就不建议培训，没啥用，想办法找个实习更靠谱）
- 3、还是学生，1w 的学费，肯定还是父母付的 。大学的学费已经让父母垫付了，还让父母支付培训费用，你好意思 ？
- 4、培训机构无下限，大学生还没毕业都诱导（比如，随随便便月薪过万，忽悠了太多大学生或局外人转行测试职业） 。

025 · 某某某培训机构靠谱吗 ？ 怎么选择培训机构 ？

IDO 老徐，答：

- 1、培训机构都差不多，没太大差别 。
- 2、别找从来没听过的小作坊 。
- 3、如果是个人弄的测试课程，先去了解下这个人靠不靠谱，多看看她的原创看看对你是否有用（如果每天都是各种转载文章，毫无观点的课本内容，就算了吧）。
- 4、全国测试培训机构名单，之前有整理，在我的公众号「简尚」回复 6161 获取（具体每个培训机构是否靠谱，不做评判，会咋了别人的饭碗，不太好。私信问我，可以给建议） 。
- 5、在缴费之前，1) 看内容大纲，2) 先去试听几节课，3) 看看讲课老师怎么样；都觉得靠谱，再考虑报名
- 6、实在做不了决定，可以微信咨询 IDO 老徐 。
- 7、延伸：如果你没有大专以上学历，放弃 测试职业（基本上很难找到工作，别听培训机构忽悠） 。

注：这一问，是给即将入行的新人的，也是关于培训机构的最完整解答 。欢迎转发给需要的他们 。

二、电子书·精选 100 篇

001 · 只需「21 天刻意练习」搞定测试工程师必备 SQL

这篇，是 1400 位测试工程师验证过，具备可行性的 21 天实战内容。
聚合了「21 天 SQL 打卡的作业 & 参考答案」，提供给所有软件测试从业者；

题目不是很难，零基础也可以跟着入门。如果你的工作中，每天要用到 SQL，那么这些你应该非常熟悉。

虽然简单，但对于 95% 以上的软件测试从业者，足够了；搞定这些，日常工作中的问题定位 & 查询数据 & 插入数据，都可以得心应手；

学 SQL，很简单，安装个 Mysql，弄个客户端，每天实操，刻意练习，21 天搞定；

每位同学建议循环 3 个周期（熟悉 -> 回顾 -> 巩固，21 * 3，2 个月）

IDO 老徐，关于学习 SQL 的 3*21 理论：

- 1) 敲一遍这些命令：熟悉 21 天
- 2) 敲完过后，如果不回归，隔段时间，就忘了：得再回顾 21 天
- 3) 再循环一次，会发现很多命令，都很有意思，爱上她了：最后巩固 21 天

学 SQL，没诀窍，就是多敲、多实践。

每天掌握一条 SQL 语句；每天一个作业，跟着实操 21 天，工作中常见的使用，就差不多了。

建议：多敲几次命令，掌握当天内容；否则，欠的[技术债](#)会越来越多，最后放弃学习。

补充，
这个玩法的逻辑是：每天一个作业，第 2 天同步前一天的参考答案（仅限参考，不一定完全正确）；

开始前的准备事项，第 0 天，

开始之前的，前置准备事项：

- 1、在自己电脑上安装一个 mysql 数据库，文章见

-> <http://istester.com/article/258.html>

2、找一个 mysql 客户端链接工具：推荐 workbench ，
安装包 <http://istester.com/article/405.html>

第 1 天作业，

- 1) 创建数据库 istester
- 2) 查看数据库创建是否成功

/

第 2 天作业，

在昨天创建的 isTester 数据库下，创建 表 istester 和 表 idoxu

注：SQL 的作业，前几天的 库、表，都是打基础；没有这些，后面的增删改查，你也没法玩；

1) isTester 表的字段、类型、属性：

id INT(10) NOT NULL UNIQUE PRIMARY KEY ,
uname VARCHAR(20) NOT NULL ,
sex VARCHAR(4) ,
birth YEAR,
department VARCHAR(20) ,
address VARCHAR(50) ,
idoxu VARCHAR(20)

2) idoxu 表的字段、类型、属性：

id INT(10) NOT NULL UNIQUE PRIMARY KEY AUTO_INCREMENT ,
stu_id INT(10) NOT NULL ,
c_name VARCHAR(20) ,
istester VARCHAR(50) ,
grade INT(10)

前一天作业答案参考，

- 1) 创建数据库 istester
create DATABASE istester;
- 2) 查看数据库创建是否成功
show DATABASES;

/

第3天作业，

根据已有的表 istester 和 idoxu 创建新表（复制表）

- 1) 复制 istester 表结构，创建新表 istester2
注：可以有两种方式
- 2) 复制 idoxu 表结构和数据，创建新表 idoxu2
- 3) 复制 idoxu 表结构的 id,stu_id,istester 三个字段，创建新表 idoxu3

前一天作业答案参考，

创建 istester 和 idoxu 表

```
CREATE TABLE istester (  
id INT(10) NOT NULL UNIQUE PRIMARY KEY ,  
uname VARCHAR(20) NOT NULL ,  
sex VARCHAR(4) ,  
birth YEAR,  
department VARCHAR(20) ,  
address VARCHAR(50) ,  
idoxu VARCHAR(20)  
);
```

```
CREATE TABLE idoxu (  
id INT(10) NOT NULL UNIQUE PRIMARY KEY AUTO_INCREMENT ,  
stu_id INT(10) NOT NULL ,  
c_name VARCHAR(20) ,  
istester VARCHAR(50) ,  
grade INT(10)
```

);

/

第 4 天作业 ,

- 1) 删除表 istester2
- 2) 同时删除表 idoxu2 和 idoxu3
- 3) 查看还有哪些表

前一天作业答案参考 ,

- 1) 复制 istester 表结构, 创建新表 istester2

注: 可以有两种方式

create table istester2 like istester ;

或

create table idoxu2 as select * from idoxu where 2=1;

- 2) 复制 idoxu 表结构和数据, 创建新表 idoxu3

create table idoxu3 as select * from idoxu

- 3) 复制 idoxu 表结构的 id,stu_id,istester 三个字段, 创建新表 idoxu3

create table idoxu3 as select id,stu_id,istester from idoxu where 1<>1;

/

第 5 天作业 ,

- 1) 修改 istester 表, 添加一个字段 istester6 (字符类型 VARCHAR, 长度 100, 不允许为空)
- 2) 查看字段 istester6 , 是否添加成功
- 3) 修改 istester 表, 删除字段 istester6
- 4) 确认字段 istester6 , 是否删除成功

前一天作业答案参考，

1) 删除表 istester2

drop table istester2 ;

2) 同时删除表 idoxu2 和 idoxu3

drop table idoxu2, idoxu3 ;

3) 查看还有哪些表

show tables;

/

第 6 天作业，

插入 insert:

1) 向表 istester 插入数据

id = 1 ,uname = idoxu ,idoxu = 2020

id = 2 ,uname = idoxu2 ,idoxu = 2020

id=11,uname = idoxu3,sex=1

id=12,uname = idoxu4,sex=2

id=13,uname = idoxu5,sex=1

id=14,uname = idoxu6,sex=2

2) 向表 idoxu 插入数据

id=4,stu_id=11,c_name=idoxu,grade=90

id=5,stu_id=12,c_name=lin,grade=100

id=6,stu_id=33,c_name=istester,grade=20

前一天作业答案参考，

1) 修改 istester 表，添加一个字段 istester6 （字符类型 VARCHAR，长度 100，不允许为空）

ALTER table istester add column istester6 VARCHAR(100) NOT NULL;

2) 查看字段 istester6 ，是否添加成功

desc istester

3) 修改 istester 表, 删除字段 istester6

```
ALTER table istester drop istester8
```

4) 确认字段 istester6 , 是否删除成功

```
desc istester
```

/

第 7 天作业 ,

1) 查询 istester 表 id = 1 的内容

2) 查询 idoxu 表 grade=100 的内容

前一天作业答案参考 ,

插入 insert:

1) 向表 istester 插入数据

```
id = 1 ,uname = idoxu ,idoxu = 2020
```

```
id = 2 ,uname = idoxu2 ,idoxu = 2020
```

```
id=11,uname = idoxu3,sex=1
```

```
id=12,uname = idoxu4,sex=2
```

```
id=13,uname = idoxu5,sex=1
```

```
id=14,uname = idoxu6,sex=2
```

2) 向表 idoxu 插入数据

```
id=4,stu_id=11,c_name=idoxu,grade=90
```

```
id=5,stu_id=12,c_name=lin,grade=100
```

```
id=6,stu_id=33,c_name=istester,grade=20
```

参考语句:

```
insert into istester(id,uname,idoxu) values(1,"idoxu",2020);
```

```
insert into istester(id,uname,idoxu) values(2,"idoxu2",2020);
```

```
insert
istester(id,uname,sex) values(11,"idoxu3",1),(12,"idoxu4",2),(13,"idoxu5",1),(14,"idoxu
6",2);
```

```
insert                                                    into  
idoxu(id,stu_id,c_name,grade) values(4,11,"idoxu",90),(5,12,"lin",100),(6,33,"istester",  
20);
```

/

第 8 天作业 ,

继续 select

- 1) 查找 idoxu 表, 名称 (c_name) 包含 “i” 的数据
- 2) 查找 istester 表, id 包含 “1” 的数据
- 3) 查找 istester 表, id 包含 “1” 的数据, 按 id 降序
- 4) 查找 istester 表, id 包含 “1” 的数据 , 取 id 最大的三个

前一天作业答案参考 ,

- 1) 查询 istester 表 id = 1 的内容

```
select * from istester where id = 1;
```

- 2) 查询 idoxu 表 grade=100 的内容

```
select * from idoxu where grade =100;
```

/

第 9 天作业 ,

- 1) 找出 idoxu 表中, 分数最高的同学和分数
- 2) 找出 idoxu 表中, 分数最低的同学和分数

前一天作业答案参考 ,

- 1) 查找 idoxu 表, 名称 (c_name) 包含 “i” 的数据

```
select * from idoxu where c_name like '%i%';
```

- 2) 查找 istester 表, id 包含 “1” 的数据

```
select * from istester where id like '%1%';
```

3) 查找 istester 表, id 包含 “1” 的数据, 按 id 降序

select * from istester where id like '%1%' order by id desc ;

4) 查找 istester 表, id 包含 “1” 的数据, 取 id 最大的三个

select * from istester where id like '%1%' order by id desc limit 3 ;

/

第 10 天作业 ,

-- 1) 找出 istester 表, sex 为空的的数据;

-- 2) 更新 istester 表, 把 sex 为空的, 设置为 0 (性别未知)

-- 3) 找出 idoxu 表, grade 小于 60 分的同学

-- 4) 更新 idoxu 表, 把 grade 小于 60 分的同学, 一律改为 59 分

前一天作业答案参考 ,

-- 1) 找出 idoxu 表中, 分数最高的同学和分数;

SELECT c_name,grade as "maxvalue" from idoxu WHERE grade in (select MAX(grade) from idoxu);

-- 2) 找出 idoxu 表中, 分数最低的同学和分数;

SELECT c_name,grade as "minvalue" from idoxu WHERE grade in (SELECT MIN(grade) from idoxu);

/

第 11 天作业 ,

-- 1) 查找 istester 表, 按 id 降序

-- 2) 查找 idoxu 表, 按 grade 升序

前一天作业答案参考，

-- 1) 找出 istester 表，sex 为空的的数据；

select * from istester where sex is null ;

-- 2) 更新 istester 表，sex 为空的，设置为 0（性别未知）

update istester set sex = 0 where sex is null ;

-- 3) 找出 idoxu 表，grade 小于 60 分的同学

select * from idoxu where grade < 60 ;

-- 4) 更新 idoxu 表，把 grade 小于 60 分的同学，一律改为 59 分

update idoxu set grade = 59 where grade < 60 ;

/

第 12 天作业，

1) 查询 istester 表，有多少条数据

2) 查询 istester 表，有几种性别类型（sex 字段，去重）

前一天作业答案参考，

-- 1) 查找 istester 表，按 id 降序

select * from istester order by id desc;

-- 2) 查找 idoxu 表，按 grade 升序

select * from idoxu order by grade;

/

第 13 天作业，

1) 查找 idoxu 表，学生成绩(grade)总分

2) 查找 idoxu 表, 学生成绩(grade) 平均分

前一天作业答案参考,

1) 查询 istester 表, 有多少行数据

select COUNT(sex) from istester;

2) 查询 istester 表, 有性别类型数量 (sex 字段, 去重)

select COUNT(DISTINCT sex) from istester;

/

第 14 天作业,

1) 查找 idoxu 表, 成绩在 80 - 100 区间的学生;

2) 查找 istester 表, id 为 2, 11, 12 的数据;

前一天作业答案参考,

1) 查找 idoxu 表, 学生成绩(grade) 总分

select sum(grade) as sumgrade from idoxu ;

2) 查找 idoxu 表, 学生成绩(grade) 平均分

select avg(grade) as avggrade from idoxu ;

/

第 15 天作业,

1) 删除 istester 表, id 大于 12 的数据;

2) 删除 idoxu 表, 分数 grade 不及格 (小于 60 分) 的数据;

前一天作业答案参考,

-- 1) 查找 idoxu 表, 成绩在 80 - 100 区间的学生;

```
select * from idoxu where grade between 80 and 100;  
-- 2) 查找 istester 表, id 为 2, 11, 12 的数据 ;  
select * from istester where id in (2,11,12);
```

/

第 16 天作业 ,

1) 造数据 , 把 istester 表的所有数据, 插入到 idoxu 表
字段取值规则
id 取 id
stu_id 取 id
c_name 取 uname
istester 和 grade 字段, 给默认值 60

前一天作业答案参考 ,

1) 删除 istester 表, id 大于 12 的数据 ;
delete from istester where id > 12;

2) 删除 idoxu 表, 分数 grade 不及格 (小于 60 分) 的数据 ;
delete from idoxu where grade < 60 ;

/

第 17 天作业 ,

-- 1) 把 idoxu 表, 改名为 idoxu6

-- 2) 检查是否修改成功

-- 3) 修改表 idoxu, 把字段 istester , 改为 istester6 (字符类型 varchar, 长度 160)

-- 4) 检查是否修改成功

前一天作业答案参考 ,

造数据，把 istester 表的所有数据，插入到 idoxu 表
字段关系

id 取 id

stu_id 取 id

c_name 取 uname

istester 和 grade 字段，给默认值 60

```
insert into idoxu(id,stu_id,c_name,istester,grade) select id,id,uname,60,60 from istester ;
```

/

第 18 天作业，

- 1) 创建数据库 istesterdb6
- 2) 检查数据库是否创建成功；
- 3) 进入 istesterdb6 库
- 4) 在数据库 istesterdb6，创建 idoxu 表，直接拷贝 istester 库 idoxu6 表的数据和结构；
- 5) 检查表是否创建成功；

前一天作业答案参考，

-- 1) 把 idoxu 表，改名为 idoxu6

-- ALTER TABLE 旧表名 RENAME TO 新表名；

ALTER TABLE idoxu RENAME TO idoxu6;

-- 2) 检查是否修改成功

show tables;

-- 3) 修改表 idoxu6, 把字段 istester，改为 istester6 (字符类型 varchar, 长度 160)

alter table idoxu6 CHANGE istester istester6 varchar(160);

-- 4) 检查是否修改成功

desc idoxu6;

/

第 19 天作业 ,

idoxu 表, 分数 grade

- 1) 排名前三的学生 和分数 ;
- 2) 排名 3 - 6 名的学生 和分数 ;
- 3) 排名 6 名以后的所有学生 和分数 ;

前一天作业答案参考 ,

- 1) 创建数据库 istesterdb6

create database istesterdb6 ;

- 2) 检查数据库是否创建成功 ;

show databases ;

- 3) 进入 istesterdb6 库

use istesterdb6 ;

- 4) 在数据库 istesterdb6 , 创建 idoxu 表, 直接拷贝 istester 库 idoxu6 表的数据和结构 ;

create table istesterdb6.idoxu as SELECT * from istester.idoxu6;

或

create table idoxu as SELECT * from istester.idoxu6;

- 5) 检查表是否创建成功 ;

show tables;

/

第 20 天作业 ,

多表 (左链接、右链接、内链接)

表 idoxu6 和 表 istester

前一天作业答案参考，

-- 1) 排名前三的学生 和分数；

```
SELECT c_name,grade FROM idoxu ORDER BY grade DESC LIMIT 3;
```

-- 2) 排名 3 - 6 名的学生 和分数；

```
SELECT c_name,grade FROM idoxu ORDER BY grade DESC LIMIT 2,4;
```

-- 3) 排名 6 名以后的所有学生 和分数；

```
SELECT c_name,grade FROM idoxu ORDER BY grade DESC LIMIT 5,666;
```

注：这题目的目的是 limit 的用法，很多同学写的太复杂；

limit 是 mysql 的语法

```
select * from table limit m,n
```

其中 m 是指记录开始的 index，从 0 开始，表示第一条记录

n 是指从第 m+1 条开始，取 n 条。

```
select * from tablename limit 2,4
```

即取出第 3 条至第 6 条，4 条记录

/

第 21 天作业（最后 1 天，结业），

1) 把表 idoxu6，改完 idoxu

2) 删除库 istesterdb6

3) 删除 istester 表

4) 清空 idoxu 表的数据

End，一切恢复如初，坐等 SQL 第 2 期；

前一天作业答案参考，

注：主要目的是熟悉这三种玩法，以及观察结果差异；

#左连接

```
SELECT * from idoxu6 a LEFT JOIN istester b on a.stu_id=b.id;
```

#内连接

```
SELECT * from idoxu6 a INNER JOIN istester b on a.stu_id=b.id;
```

#右连接

```
SELECT * from idoxu6 a RIGHT JOIN istester b on a.stu_id=b.id;
```

/

结业，最后一天作业答案参考

1) 把表 idoxu6，改名为 idoxu

```
rename table idoxu6 to idoxu;
```

或

```
alter table idoxu6 rename idoxu;
```

2) 删除库 istesterdb6

```
drop database istesterdb6;
```

3) 删除 istester 表

```
drop table istester;
```

4) 清空 idoxu 表的数据

```
truncate table idoxu;
```

或

```
delete from idoxu;
```

End，恭喜完成一轮刻意练习。

循环 3 次 21 天打卡，总共 63 天（两个月掌握日常工作中，测试工程师必备的 SQL）

IDO 老徐

2022.04.27 深圳

#精选 100 篇#

002 · 互联网公司，一名「测试工程师的一天」

副标题：软件测试工程师的一天，时间节点？工作内容是什么？每天在忙什么？

作者：IDO 老徐

8:00

被连着 3 个闹钟，从睡梦中催醒。

9:00

在床上磨蹭了一个小时，被最后一个闹钟惊醒；一看 9 点了，马上要迟到了；慌忙洗漱，一路狂奔到公交站 / 地铁站。

09:30 - 10:00

下地铁，一路狂奔，踩点到了公司，打卡，打开电脑，看着回不完的群消息。例行早会，聊聊昨天干了啥、今天干点啥、有啥棘手的问题、以及需要跨部门 / 跨团队协调的事。

11:00

- 1、打开昨天没测完的系统，写测试脚本、执行用例、测 Bug、验 Bug、ReOpen Bug。
- 2、催开发、怼产品、看简历、面试。
- 3、打开邮箱。
- 4、打开 BUG 系统。
- 5、打开各种每天必用的软件（测试管理工具，接口测试工具，wiki 等）。

12:00

一晃，到点、吃饭，找三五同事，吃饭聊天、逛 [isTester](https://isTester.com)、刷抖音、补个觉。

13:30

被同事们的劈啦啪啦的键盘声 & 激烈的争斗声，吵醒。
半睡状态，磨蹭，一晃，14 点。

14:00 - 16:00

重复 11:00 - 12:00 的工作。

17:00

快下班了，坐等即将提测的新版本 / 即将召开的需求评审会议。

18:00

快到下班点了，瞅一瞅今天的任务，还有一大半没完成。
收到新版本提测邮件了。

19:00

找几个同事，一起吃晚饭。

20:00 - 21:00

吃完晚饭，回到公司，开始测试今天提测的新版本。

22:00

工作间隙，开始了一天的[学习时间](#)。

各种脚本语言、[Linux](#)、Jenkins、[SQL](#)、[Git](#) 等，学起来，很开心，毕竟有一堆这个时候[正在学习的同伴](#)。

23:00

下班了，打车回家（车费公司报销，又免费蹭了一次打车，每月总有那么 20 天蹭车的机会）。

00:00

到家，洗漱，刷抖音，睡觉。

02:00

还在刷抖音，真的要睡觉了，明早起不来了。

06:00

响了第 1 个起床闹铃，沉睡的你，并没有听到。

08:00

重复着，美好的一天 ...

未完待续 ...

这是 V2.0 版本（V1.0 版本，首发 2020 年）

注：如上为主线，还有一些可能性的支线。

1、在 02:00 - 08:00，你一直在公司忙的通宵加班。早上 6 点，保洁阿姨又一次跟你打了声招呼。

2、在 19:00，好像没什么事，你偷偷的下班了，刚到家，被 Leader 一个电话，原路返回。

3、在深夜 02:00，系统出 Bug 了，一声惊醒，继续点点点。

4、这周啥事没有，假装很忙的坐在工位上，学习者「[21 天 Linux 打卡命令](#)」

5、从早上 10:00 - 18:00，一直在开 [需求评审会议](#)，主题越来越扩散，你强忍着，让自己不睡着。

6、穿梭在各种[用户问题反馈群](#)，解答着各种神奇的问题。用你的专业知识，排除可能性，定位 Bug。

7、又摸鱼了一天，苦思今天的日报，到底要写点啥。

8、早上到公司，突然收到一封老板告全员信，公司破产清算了。

9、正在聚精会神的提 Bug，突然被人事叫去，告诉你，今天毕业了（被裁员了），拿着 N+1，从此自由人。

End。

IDO 老徐

2022.04.27 深圳

#精选 100 篇#

003 · 一个测试经理的自述：我是如何管理测试团队的

对于刚从测试人员转向测试管理岗的同学，肯定会有很多疑惑，不知如何下手。且一时观念无法转变，到底该如何管理测试团队？

关于管理，行业内，有太多专题文章

这篇文章，IDO 老徐主要分享自己的经验，以及老徐是如何管理测试团队的

注：仅个人经验分享，供参考。

— 正文 —

测试管理，范围很广

带 1-2 人也是管理

带几十人也是管理

但是管理方法肯定会不一样

这篇文章，以 100 人左右的测试团队为例，聊聊是如何管理的

1.

首先，根据业务情况，或者项目情况，拆分成几个测试小组；

每个组，有一个测试负责人（测试组长）

IDO 老徐只需直接管理每个组的负责人（测试组长）就行。

(注：每个人的最佳管理人数是有限的)

2.

充分的放权

各小组负责人对各自负责的小组质量负责

涉及各小组配合的项目，存在边界问题，可随时找我沟通

而且老徐也会每周进行 1-2 次的分享

以及每周一次的周会沟通

以及，每周所有成员的周报

3.

问题分析

老徐喜欢分析漏测的问题

对于各类漏测的问题，重点分析

对于测试同学，这是最快的成长方式（分享失败经验，从失败中吸取经验是最快的成长方式，避免重复出现类似问题）

每个测试同学，工作过程中都会出现漏测的情况，发布出去的版本存在各种质量问题，但是如果同类重复多次出现，得反思。

4.

充分的自由

老徐只关注结果，所有测试同学对结果负责

如果能高效高质的完成工作

老徐不太会关注你工作是否摸鱼 。

如：上班偶尔聊聊天，早上晚点到，临时有事提前走几个小时，都 ok

(前提：管理者所在公司给你放权，你有说话权；很多公司管理非常严格，老徐不太赞成那种工厂式的管理方式)

5.

自动化保障、内容沉淀

所有的接口、构建、部署，全部自动化实现

对人的依赖性小 。

所有资料，全部 wiki 在线写，不写线下文档（避免离职交接的问题） 。

6.

招聘、人员补充

根据当前团队成员情况、以及产品项目情况，招聘互补型的成员

对于测试管理者，首页必须要了解团队中每个成员的详情情况

每个人的优势、性格、擅长测试的类型

IDO 老徐

2022.04.27 ， 基于 2016 年的文章 二次写作

004 · 避免试用期，被辞退，指南 01（高级岗位）

这篇文章，主要针对软件测试从业者，列出具体建议。

根据从业者，所处的阶段，分三篇文章来写。

这是第一篇，**给高级岗位的同学**

第二篇，会针对中级从业者

第三篇，会针对刚入行的职场小白（菜鸟，刚入职或转行的新人）

1. 开始之前，先定义几个关键词。

高级岗位：部门 Leader，或者团队 Leader，或者核心测试开发岗位（至少是在某个领域，要有独立产出的）。

中级岗位：能独立负责一个项目的整体测试工作，常规来看，2-3 年的从业者。

初级岗位：刚入职，或者入职 1 年左右。

2. 基于如上前提，达成共识后，开始此文。

高级岗位，有几种情况。

- 1) 你的上级是质量部 Leader（经理/总监），或者技术 VP 等
- 2) 你的上级是公司老板

3. 如果你的汇报对象，是质量部 Leader，

1) 你应该思考下，如何根据部门当前现状，当前最严重的 Top 3 问题或事项，给出你的解决建议。

2) 先跟部门 Leader 沟通，了解下他对你的期望，过来解决什么问题，文档化（或者思考脑图），产出你的整体规划（至少给人感觉，你的经验是丰富的、逻辑思维是清晰的、态度是认真的），剩下的就是当面碰，得出优先级高的 Top3

3) 剩下的就是落地能力，以及事项推进能力（主动去推进一些事项的落地、主动汇报进度、主动抛出问题并给出一些你的思考），这个岗位，不是单纯的执行岗位，切勿等着 Leader 给你逐条指派任务（这种同学，很危险，也许突然就会被辞退了）。

4) 万一遇到，老大对你某些内容不太满意，没有达到他的期望，他会找你沟通，把沟通的问题，都记下来，一一去高效的解决掉（至少让 Leader 感觉，哟，不错哦，轻轻点播，即有意外惊喜，可教也）。

5) 还有就是日报、周报、文档输出等软技能，也是非常重要，文章底部统一讲解。

4. 如果你的汇报对象，是老板（不懂技术的老板。懂技术的老板，也不关注你的技术细节）

- 1) 你应该思考下，如何规划部门的发展方向，提升全公司的研发效率 & 质量（减少线上问题反馈量，以及 Bug 漏测率）
- 2) 制定一些流程 & 规范，来跟老板一些安全感（发出去的版本，一定没有问题，就算有问题，你也可以搞定）
- 3) 目前公司存在的一些问题，思维导图梳理出来，并给出你的一些建议。
- 4) 如何来培养团队，提升团队的能力 & 士气。
- 5) 通过实际的数据来体现你的价值，比如（Bug 率每周下降，把其他部门关注的数

据，都报表化，每周同步给相关部门等）

如上，这些，都是应该思考的。

5. 怎么写日报、周报，如何输出大家需要的文档？

作为高级岗位，文档是最基本的能力，你的价值，除了自己产出一些内容，更重要的是如何把这些能力，快速复制给更多同事，或者其他部门。

这个时候，就通过文档的形式来体现了（所以，平时多写写文章，间接对工作，还是有点帮助的）。

写文档，几个点：

- 1) 层次清晰，阅读者快速知道，文档上，哪些对自己有用，自己需要关注哪些点（根据角色来分布 或者 重点内容特色颜色标记）。
- 2) 工作文档，别写废话，直接写工作中能落地的，拿来即用（很多同学，喜欢从网上抄一堆介绍，没啥用）
- 3) 一个好的工具，Wiki 不错，可以了解下（实在不行，Git / svn 都可）

日报，

今天完成了哪些有价值的内容，进度多少，明天计划做些啥，有哪些问题需要协助的。

就是这么简单。

很多同学，也许没写过日报。

这里给个建议，**试用期员工，必须写日报，好及时了解问题 & 能力。**

周报，

同上，只是范围，从日，到周。

End，

搞定如上几点，平稳度过使用试用期，完全没问题。

一般来说，试用期搞定后，后续被辞退的可能性，就更低了（企业重新招聘一个人的成本，太高）。

很多时候，能力不一定是最大的瓶颈。

能力不够，可以随时恶补。反而，态度、工作习惯，是很难短时间改正的。而且，这些，也是面试时，很难暴露出来的。

这是第一篇，针对高级岗位。

如上，IDO 老徐，一家之言，你有任何疑问，欢迎留言讨论。

希望，此文，对你，有那么点，价值。

注：这篇文章，建议反复阅读 10 遍，并记下来，关键时刻用得上。

IDO 老徐

2022.04.27，最早写于 2018 年，略有调整。

005 · 避免试用期，被辞退，指南 02（中级岗位）

对于中级岗位，大体框架，共 10 点

注：中级岗位，才是数量最庞大的群体，这篇文章，建议多看几遍。

1. 快速熟悉公司整体业务

作为一个合格的中级岗位，一个合格的测试工程师，你得独立负责某个项目/业务的质量工作（至少也得独立负责某几个模块的业务测试，或者自动化测试的执行工作），相关的业务线知识，必须得了解。一般来说，一个公司的任何业务，必定是与其他业务有关联的。

熟悉后，可以快速定位，问题来源，问题原因，每个问题知道找哪些同事协助，提升工作效率。

延伸，

如何快速熟悉业务？

看 Wiki 文档、看用例、看 Bug、看线上反馈的问题、问测试同事、问相关业务线开发、问产品经理、自己把系统完整的玩几遍。

2. 快速了解线上线下 Bug 提交、跟进套路

跟进 Bug，是测试必备的基础能力，不用多说。

比如，

跟进流程是怎样的？各种问题如何解决？需要录入到系统否？（每个公司不同，进入公司后，得快速了解这些。一般来说，公司都有文档说明，或者有导师带你了解这些）

3. 快速了解日常工作所需工具

比如，

链接 Linux 服务器的工具 Xshell 等，链接数据库的工具 Navicat 等，Git 客户端，Bug 管理工具禅道、Jira 等，用例管理工具，日常文档协作工具 Wiki 等，很多很多。

这些，至少一周内得搞定。

职场没太多时间让你慢慢熟悉，这个阶段，大家都不是新人，招你过来，就是快速干活的。

延伸，

如何能快速掌握？

平时工作之余，就得掌握这些工具的日常使用（少抱怨、少闲扯、少玩游戏）。

去任何公司，只是规范不同，工具使用无门槛，几天可搞定。

4. 快速跟部门内同学混熟

不用多说，如何与人沟通，职场必备软技能。

5. 快速跟工作需要协调的相关人混熟

开发、产品、设计、客服 等。

不用多说，如何与人沟通，职场必备软技能。

6. 日事日清

把每天老大交付的任务做完、做好，且反馈结果

7. 及时响应

交付的工作任务，及时响应，有结果后，及时同步（即使没完成，遇到困难，也同步进度，知道你在干这事）

8. 结果输出

测试结束了，及时产出测试报告。

用例完成后，及时输出文档（不管是 Excel 形式、还是走的用例管理系统），安排评审（看公司具体流程，有些需要 2 次评审，有些公司无评审环节）

9. 让 Leader 放心

每次的任务，都能主动思考，交付出满意的结果，或者微创新，Leader 会对你刮目相看。

10. 文档落地、沉淀（不仅自己会，还得让其他同学会）

养成输出文档的习惯，一个人的研究成果，可以让多人快速掌握，不用重复投入时间。这是 Leader 比较喜欢的员工。

如果真能做到如上几点，至少超过 80% 的「同阶段从业者」。

过试用期？肯定没啥问题。

而且，还会非常优秀，假以时日，可成长为一名合格的高级测试。

搞定如上 10 点，平稳度过试用期，完全没问题。

一般来说，试用期搞定后，后续被辞退的可能性，就非常低了。

希望，此文，对大家，有那么点，价值。

IDO 老徐

2022.04.27，最早写于 2018 年，略有调整。

006 · 避免试用期，被辞退，指南 03（初级岗位）

其实啊，想度过试用期，挺简单的。

企业招聘人才的成本挺高的，既然能成功入职，至少说明，符合岗位需求的。只要不是培训机构包装的「刚毕业，伪装 3 年经验」，或者「工作了两年，却啥都不懂，还不如应届生」，就不会有太大问题。

每个岗位，都有对应的预期。对于初级岗位，Leader 对你的预期也不会太高。期望不高，失望就不会太大。

初级岗位，避免被辞退，总结了，就 17 个字：「态度好、好学习、不犯原则性错误、知错能改」

17 字拆解如下：

1.认真看看公司的行政规范，别触发公司原则性问题。比如，迟到、早退、虚假学历、虚假工作经验等。

2. 把新人入职指引上面的内容，尽快弄懂
一般，有两块指引：行政这块的入职指引 和 部门这块的工作指引。

比如，

- 1) 日常工作所需工具：Bug 工具、用例工具、代码管理工具、项目管理工具、Linux 客户端工具、Mysql 客户端工具等；
不会的自己百度，百度后，折腾了几小时还是搞不定的，问下同事，别再瞎浪费时间了。也许是你折腾的姿势不对。
- 2) 公司的组织架构情况，了解下哪些部门，与哪些部门需要强协作的。部门的同事，研发同事，产品同事等。

3. 学会提问

- 1) 做一件事，不确定具体的目标之前，先跟老大确认下。别折腾几天，最终做的结果，不是 Leader 需要的。同类事情，发生几次，Leader 就没有耐心了。
- 2) 你工作中，遇到的所有问题，同事都已经遇到过。汇总自己不懂的问题（千万别一个个问，一天问 N 次，多了，谁都烦），问身边的同事。或者，问你的导师（很多公司，对于入职的新人，会安排一位导师，全程解决，你的各方面问题，好好利用这块的资源福利）。

4. 看看历史 Bug 库、用例库

便于快速了解系统现状，存在的问题，通过问题，快速熟悉系统。特别是你即将要负责的业务模块，一定要弄懂。

5. 有时间，多用用自己的公司产品。
技术不足，业务来补。

6. 认真完成，Leader 交付的每个任务，完成后，反馈结果。
就是这么简单，初级岗位，多数都是执行类事物。执行完成后，同步结果即可。

7. 充满激情。
激情很重要，毕竟年轻（这条好像也没啥实质性作用）。

搞定如上，平稳度过试用期，完全没问题。

一般来说，试用期搞定后，后续被辞退的可能性，就更低了。

毕竟，经过试用期三个月培养，已经差不多熟悉业务，可干活。且招聘成本那么高，重新招聘，成本太大，入职周期也是未知。另，试用期过后，辞退，涉及赔偿事宜，麻烦。

很多时候，能力不一定是最大的瓶颈。
能力不够，可以随时恶补。反而，态度、工作习惯，是很难短时间改正的。而且，这些，也是面试时，很难暴露出来的。

这是第 3 篇，针对初级岗位。
至此，试用期系列，3 篇，完整交付，希望对大家有点帮助。

希望，此文，对大家，有那么点，价值。

IDO 老徐
2022.04.27，最早写于 2018 年，略有调整。

007 · 我经历的 5 种测试组织模式

常见的，这 5 种测试组织模式，

- 1、独立部门：整个公司，有独立测试部门，所有测试人员，归属于这个测试部门，汇报：测试经理/测试总监。
- 2、分散模式：没有独立的测试部门，测试人员，分散在每个独立的业务线，每个业务线是一个「测试组」，汇报：测试组长
- 3、混合模式：有产品测试部门（类似 saas 平台研发，基础产品测试），其他业务线或者定制项目，每个业务线/项目，有一个独立的「测试组」，汇报：测试经理/测试总监 及 测试组长 / 开发组长。
- 4、外包第三方模式：公司内部不招测试工程师（或，以外包测试为主），把这件事交给专业的第三方外包公司，由他们派人驻场测试（是的，很多所谓的外包，就是这种玩法）。汇报：项目经理 / PMO

5、无测试模式：不需要全职测试工程师；或者每个项目团队，自己招聘几个测试工程师，与开发工程师，密切合作，组成项目组，向开发组长汇报；没有所谓的「测试组 / 测试部门」，没有测试 Leader 岗位。
人人都是测试工程师，项目团队，一起为质量负责。用技术手段、流程辅助 等。

@IDO 老徐，注：

从过往的 数据调查报告、过往 6 期 百人计划的问卷数据 及 这些年与数万名测试工程师的交流，大体如上 5 类 组织模式。

互动：你公司属于哪种？评论区 回复 对应序号。

如果不包含在 如上 5 种，欢迎评论区 聊一聊，一起学习 下，感谢。

延伸：哪种模式最好？

- 1、从管理者职业发展角度，1 或 3 更佳，
- 2、从企业效率角度，各有利弊，
- 3、从个人角度，尽量别去 4（外包）

IDO 老徐

2022.04.27，最早写于 2021 年

008 · 软件测试之外包利弊

“企业，永远在追求利益最大化。不以盈利为目的的公司，都是耍流氓。” -- IDO 老徐

从利益最大化的角度。一些企业，不想组织结构过于臃肿；或者带来的管理麻烦。进而产生了一种新的雇佣方式：外包。

外包，分多种。比如：项目外包、人力外包。

项目外包：某个项目，全部交由第三方团队来做，总共多少钱、以及项目交付截止时间，谈好即可。时间到了，来验收项目即可。至于，具体谁来做，几个人做，不管。花钱、省心；就是所谓的甲乙双方。

人力外包：企业需要做一个项目，本部员工不够，又不打算自己重新招聘员工。与第三方人力机构合作，临时弄一堆员工过来，加入某个项目，参与一些非核心的测试需求（就是所谓的点点点工）。

项目结束，人员撤离。

企业按人头付费给第三方机构。

外包人员，虽然在企业上班，但是发工资，是第三方人力机构。

日常所谓的外包，更多是 人力外包项目外包 。
这篇文章，梳理几个人力外包的利 & 弊 。

希望，对迷茫，不知如何抉择的你，一些参考 。

首先，外包的 利：

1. 对于刚入行，以正式员工的身份进入大厂，能力还不够时，可以去外包，沉淀 1 - 2 年，积累经验 。
2. 对于一直在小作坊就职的同学，可以去外包待 1 - 2 年，了解下正规的项目流程是怎样的。对自己的后续职业发展，有益 。

其次，外包的 弊：

1. 没归属感：驻场在其他公司，你就是一个干活的，活干完，走人。下一个项目，还不知道在哪家公司呢。
在客户现场上班，却貌似又不归属于甲方。似公司人却又不在公司上班，不似甲方人却又归属于甲方管理，显得不伦不类。
2. 能力成长有限：由于多数是边缘项目，很少能接触到项目核心内容。而且，很多代码库、表、服务器等，根本没有权限查阅 。
3. 工作不稳定：很多时候，项目结束，意味着，就要重新去另外一个甲方，重新参与项目。很多时候，接不到项目，甚至会出现被裁撤的可能性。据说，有同学，亲身经历：自己所在项目组，项目一期结束，直接有部分人被清场回家了。
4. 另外，外包的工资低（当然，也有很多同学，在外包，工资还行）。而且，啥福利都没有，据说年终奖都没有 。
5. 没有话语权，就是一个执行干活的 。

最后，给一些建议，

1. 新人，实在找不到更适合的工作。可以去外包，历练 1 - 2 年 。
2. 工作多年了，就没必要去外包了。赶紧跳出来 。
3. 外包，不建议待太长时间（之前面试，接触过一些在外包待 5 - 6 年的，与外界脱离太久，基本上就废了）

关于外包的问题，看这篇就够了 。

太多同学问，太多同学讨论，避免重复解答，特意整理一篇文章 。

希望，后续不要再收到关于外包话题的重复问题了，觉得有用的，自己收藏 + 发给朋友 。

补充，

非常遗憾，十几年工作经验，老徐没有经历过外包；如上的利弊分析，都是老徐了解到的一些内容，以及一些个人分析。

供参考，
我说的，也许是错的。

IDO 老徐
2022.04.28，最早写于 2020 年

009 · 软件测试工程师 25 条可行性职业发展路径

先说观点：那些，担心 30 岁后、35 岁后、40 岁后，无路可走的、找不到工作的。基本属于能力不够，或者思维太局限，或者思考太少的。

总之，瞎担心、不长进、不执行、不落地。

为什么会有这样的观点？看完这篇文章，你就解惑了；自行收藏 + 转发，此文内容三年、五年、十年，依然不会过时。

曾经，写过一系列的软件测试从业者职业成长文章，本以为，大家应该都目标清晰，职业发展路径清晰。

但越来越多的，软件测试从业者咨询老徐，关于职业发展的疑惑。

我在思考，是有必要专题写一篇，测试从业者，可行性的发展路径了。
参考这些路径，看看哪条离你最近、最有可行性？剩下的，就是执行了。围绕这些目标，此刻开始，向目标靠齐。

道理懂再多，真到自己的身上，需要做关键决策时，依然不敢做决定、迷茫。

通过这篇文章，再次聊聊，软件测试工程师的，职业发展，可行性的发展路径；到底有哪些路可以走，分别有哪些门槛，哪些坑，适合谁等。

希望对你，有那么点用。

认准目标，剩下的，就是，执行、执行、执行。

从普遍的思维来看，测试从业者的发展路径，技术方向 & 管理方向。
但，此文，给大家提供了 第 三 方向。

我们先看 通用的两个方向，

一、技术方向

1. 测试开发工程师:

支撑测试部门，服务于测试部门；通过系列的技术手段，提升测试团队的工作效率；如开发测试部门所需的测试工具、各种脚本；开发测试部门的测试平台、写一些能提升业务测试工程师效率的造数据小工具（毕竟造数据是很多团队效率障碍的真凶之一）；

偶尔，也需要 做 性能测试 。

当然，不同公司不一样。不排除，有些公司这个岗位，让你去测功能偏多；或者利用已有的测试工具，执行自动化测试 & 性能测试 等。

门槛：必须具备一定的代码能力（能看懂、能写）；最好有一点测试思维，能理解测试部门的业务需求，对测试整个流程、诉求、痛点了解，能基于诉求、痛点，开发出能解决问题的小工具，提高效率。而不是炫技，搞一堆「华而不实」的无工具，自嗨。还在那苦恼，为什么你写的工具测试团队不去用？

问问自己：你开发的工具，不仅解决不了问题，还给业务测试增加工作量，为什么用？

2. 自动化测试工程师：通过开源的自动化工具，或者一些商业工具（或者公司内部测试开发团队写的工具），执行自动化测试，以及确定自动化测试方案等。

门槛：对常用测试工具熟悉，了解自动化测试流程，得具备基本的代码能力等，能进行一些简单的脚本修改 & 编写。

延伸，

自动化测试工程师，现在其实也归类到「业务测试工程师」的范畴了，见第5项。

3. 性能测试工程师：不仅仅是用 Jmeter 等工具来测试性能；而是性能分析、风险控制、性能评估、给出调优建议 & 结果等。

门槛：水很深，不是玩玩工具就可以的。IDO 老徐不建议，各位走这条专项路径。且专职岗位不多；想专项玩，门槛高；

4. 测试架构师，整个测试部，技术方向把控。

门槛：测试方面的理论、方向，软件工程，语言基础，系统架构 等，都必须得具备；且需要一定的沟通能力 。

延伸，
测试架构师 的 误区、现状、能力要求

- 1) 很多公司，是没有这个岗位的
- 2) 是测试团队，测试技术 & 测试经验 的领军人物，具备团队影响力
- 3) 新测试技术、方案、工具 的预研；引入新的测试技术，及 测试方案，带领测试部门，高效高质
- 4) 培养公司的测试工程师 梯队
- 5) 测试架构师，需要有非常强的沟通能力；否则，很难在测试团队形成影响力（是的，光有技术硬技能，是不够的）
- 6) 需要对公司的业务，及团队的整体技术情况有了解，否则新技术方案，无法落地
- 7) 对整个测试行业的发展 及 最新实战 了解，综合评估是否可引入
- 8) 测试架构师，不是测试开发工程师，也不是测试 Leader（国内兼任的情况，除外）

问：你公司有「测试架构师」角色 否？

5. 业务测试工程师（title 也许是：测试工程师 / 高级测试工程师 / 资深测试工程师 / 自动化测试）：一般来说，也是技术方向，得专项到某个业务方向，偏业务方面多点 。

门槛：稍微有点技术能力，不是点点点；具备基础的测试必备能力，linux/sql/Shell/Python/Java/工具等；当然，很多不会的也混的很好，只是核心竞争力不够 。

6.补充，

- 1) 安全测试 方向，忽略 。多数测试从业者，不具备这个能力，做不了 。一般是交给第三方团队来做 。
- 2) 性能测试 方向，不太建议，坑深，且专项岗位不多 。

二、管理方向

1. 测试组长：公司只有几个测试的情况下，一般会有个测试组长，管几个业务测试工程师，团队功能测试为主，无专职测试开发岗位 。

2. 测试主管，同 1，只是岗位名不同；
唯一区别的是，测试主管上级可能是测试经理；测试组长上级，应该就是研发老大，属于研发部门的一个小组（当然，只是大方向，不一定完全这样；也有很多独立的测试部门，组织架构是：测试组长 -> 测试经理 -> 测试总监）。
3. 测试经理：一般来说，管理一个测试部门，下设多个「业务测试组」；也许会有一个独立的「测试开发组」服务于业务测试团队，提升效率；除了关注技术本身，还得关注部门发展、绩效等（是的，具备 升职、加薪、裁员 等权限）。具体可以看 003 篇 <https://t.zsxq.com/B6iEYjA>
4. 项目测试负责人：很多项目制形式的团队，带领几个测试人员，对这个项目的整体质量负责，协调资源，确保整个项目最终质量；参考（业务测试工程师）。
5. 测试总监：对整个公司的所有系统质量负责，质量部门的最高负责人；测试经理延伸；
市场上岗位有限，看机遇，缘分。
6. 质量副总裁（测试副总裁）：看到有一些公司在招这类岗位，但非常少；具体内容，参考 测试总监，没啥太大区别；
7. 研发总监：很多技术体系 + 技术底层沉淀足够的，且运气不错，加上自己足够努力的，转「转发总监」也是可以的，不过难（门槛高）；
8. 事业部总监：做测试的过程中，带某条业务线，深度足够后，如有机会，可以直接负责那条线，最终运气不错的，有成为事业部总监的可能性（身边有成功案例），不过看运气 + 自身努力（门槛高）；

三、其他方向：聊了两个常规方向，再带大家看几个其他方向

思维扩散点，除了 技术 或 管理，其实还有几个方向可发展；
不是非要死在测试这颗树上。

比如，

1. 产品经理：测试转产品，是非常适合的一条路，身边很多这类成功案例；毕竟，平时项目中，测试是对业务最了解的；
2. 开发工程师：参考 测试开发工程师；
3. 转行做「咨询顾问」：也挺好，很多这类独立顾问，但机会不多，看个人影响力 + 表达内容 + 知识体系 + 见识（高门槛）；

4. 转行做「培训老师」（培训机构很火，而且培训老师，技术成长压力没那么大，也有非常多成功案例的）；
5. 回家带娃（特别是职场妈妈）：老徐作为过来人（家里两娃），建议小孩尽量自己带，就算自己带不了，也要留在身边，每天陪伴；父母带与自己带，差别一点点；
6. 自己创业（工作了几年，有一定的基础资金，去创业，做点自己感兴趣的）：而且 50 岁、60 岁，找工作挺难的，创业是一个方向；身边也有案例；至于创业做什么，不重要，不一定要是 IT 相关的；
7. 转行做销售（有一定的技术功底，如果口才 OK，做销售，也是条路）
8. 转行做技术支持：售前 / 售后 也可以；应该都知道吧，很多公司有这块岗位，测试的工作经验，对技术支持岗有帮助；
9. 转行做「运维」：其实也是不错的方向，平时每天接触 Linux，自己看服务器监控、自己搭服务器、啥都会的；做做运维，混口饭吃，也没问题；
10. 转行做「运营」：运营适合脑洞大、对数据增长感兴趣、或者文笔不错的同学，也许是一条路，可以试试；
11. 找一份普通岗位的测试工作，有一份基本的薪资；然后，利用业余时间，玩副业，沉淀「个人 IP」，探索可能性，应对大环境危机 和 年龄危机 。

End

补充：经常有同学问 IDO 老徐，“软件测试从业者 35 岁以后，都干嘛去了？”看完这篇文章，你也许就明白了；

路有太多，如上这 25 条，都具备操作的可行性，且有成功的案例（说明，路径通）

选什么不重要；最重要的，是自己的兴趣，且持续努力，没有随随便便「成事」的。

IDO 老徐

2022.04.28，最早写于 2017 年，此次做了比较大的调整 和 内容补充。

010 · 21 天，21 个简历面试问题，搞定所有

注：这篇文章，建议打印出来，多看几遍，每天收到的无数问题，几乎都包含在这 21 个问题里。这 21 个问题，最早发布于 2020 年，1400 位参与了 21 天打卡的测试工程师都知道。最近在整理电子书，觉得有必要把这篇收录。

简历 & 面试 21 天打卡，第 3 期 01 天 0530

01

简历制作：

- 1) 简历一般几页？
- 2) 什么情况下需要贴照片？
- 3) 需要重点体现啥关键词？怎么体现在简历中？
- 4) 项目中，如何体现技术关键词？与简历关键词，呼应；

/

简历 & 面试 21 天打卡，第 3 期 02 天 0531

02 薪资范围估算

思考：

- 1) 怎样找到自己当前阶段的薪资范围？

避免面试时报的太高（失去 Offer）或太低（面试官觉得你不靠谱）

曾经直播分享了四种方式 <http://istester.com/article/396.html>

结合自身去试试。

前一天，参考答案：

- 1) 简历建议 2 - 3 页
- 2) 测试简历，不建议贴照片；除非是妹纸，且特别漂亮；
- 3) 测试必备的能力且自己已经掌握的，一定要体现；如 linux sql java/python git jenkins jira jmeter 等等（也可以列 15 款可尝试的接口测试工具）；
- 4) 自己简历上体现的，已经会的内容，需在「项目描述」中有体现（让面试官感知，你有实战经验）

/

简历 & 面试 21 天打卡，第 3 期 03 天 0601

03

- 1) 用一段话，描述你最近做的项目，以及负责的内容；

前一天，参考答案：

- 1) 去招聘网站搜索，了解测试的大概薪资
- 2) 询问同城市测试人员
- 3) 寻求前辈定位
- 4) 结合自己的上份薪资，上涨 20%+，以及结合面试表现（加减 20%）

/

简历 & 面试 21 天打卡，第 3 期 04 天 0602

04 前置准备

开始找工作之前，需要提前多久准备？需要准备哪些内容？

前一天，参考答案：

比如，

最近半年，负责的项目是 xxx 平台，用户群体是 xx，用户量 100W 左右，团队 20 人，16 开发、4 测试；主要工作是业务+接口，涉及工具 xxx 等；

/

简历 & 面试 21 天打卡，第 3 期 05 天 0603

05 投递简历渠道

投递简历，有哪些常用的渠道？

你一般在哪投递？为什么？

前一天，参考答案：

- 1、建议至少提前 3 个月准备
- 2、对比老徐之前写的软测知识体系，再对比自己掌握的内容，找出欠缺点，恶补；
- 3、把自己简历上的内容，彻底弄懂；
- 4、物色想去的公司，提前准备；

/

简历 & 面试 21 天打卡，第 3 期 06 天 0604

06 面试邀约筛选

收到面试电话后，需要问啥？

多个面试邀约，如何筛选？

前一天，参考答案：

渠道：boss、拉勾、51job、猎聘、内推、圈子（比如，咱们的「21 天打卡」星球，也是可以互相内推的）、微信群

/

简历 & 面试 21 天打卡，第 3 期 07 天 0605

07 自我介绍

面试，自我介绍，应该怎么介绍？

听直播前，自我介绍这块，你犯过哪些错误？

前一天，参考答案：

- 1) 测试团队、公司属于哪方面业务、研发测试比例、这个岗位的负责内容、薪资范围等；以及详细信息让发面试邮件；
- 2) 根据面试邮件 & 个人定位，选 2-3 家去面试；

/

简历 & 面试 21 天打卡，第 3 期 08 天 0606

08 面试过程

面试过程，遇到不会的问题，怎么处理（怎么回答）？

前一天，参考答案：

自我介绍：哪一年参加工作，测试工作几年，经历过哪些公司，参与过哪些项目，主要负责的内容，技能擅长等（两分钟内介绍完）

/

简历 & 面试 21 天打卡，第 3 期 09 天 0607

09 印象深刻的 Bug

面试官问：“说说你印象深刻的几个 Bug？”

从那些角度回答 ？

前一天，参考答案：

直接说这块不会，也可以大概说下你的思路和理解（注：针对每次不会的问题，面试结束后，得恶补，说明这块知识点有欠缺；面试是一个非常好的成长过程）

/

简历 & 面试 21 天打卡，第 3 期 10 天 0608

10 为什么离职

面试官问：“为什么离开上家公司（为什么离职） ？”

前一天，参考答案：

思路：

bug 的严重程度，所带来的的影响，什么情况下发生的，你是怎么定位到的，你是怎么解决的 ；

最好是哪种比较难发现的，经过各种手段，最后才找到原因的 ；

/

简历 & 面试 21 天打卡，第 3 期 11 天 0609

11 期望薪资

面试官问：“你期望的薪资 ？”

前一天，参考思路：

- 1) 个人发展、公司前景、项目解散、公司倒闭 等 ；
- 2) 别说公司的问题、别抱怨；
- 3) 喜欢当前面试的公司，期待加入；

/

简历 & 面试 21 天打卡，第 3 期 12 天 0610

12 入职后如何开展工作

面试官问：“如果给你发 Offer，你入职后会如何开展工作？”，怎么答？

前一天，参考思路：

根据 02 节定的标准，根据此次面试表现，上下浮动 20%

/

简历 & 面试 21 天打卡，第 3 期 13 天 0611

13 面试反问

面试结束后，面试官问：“你有啥想问的？”，你应该问些啥？

前一天，参考思路：

- 1) 普通岗位：了解业务、熟悉业务、看文档、缺陷库 等
- 2) 管理岗位：去发现问题、沟通了解、分析总结，给出方案，解决 TOP3 问题（团队 & 业务） 等；

/

简历 & 面试 21 天打卡，第 3 期 14 天 0612

14 多个 Offer 选择

多个 Offer，如何抉择？判断标准是啥？

前一天，参考思路：

了解 团队结构（开发测试比）、接下来半年的规划、自己负责的内容、工作流程、用到的系统 等，跟这个岗位强相关的内容；

/

简历 & 面试 21 天打卡，第 3 期 15 天 0613

15 什么时候提离职

如果打算换工作，什么时候跟现在公司说，比较好？

前一天，参考思路：

- 1) 职业发展吻合
- 2) 公司前景
- 3) 薪资
- 4) 根据你当前阶段，最关注的 TOP，对比；

/

简历 & 面试 21 天打卡，第 3 期 16 天 0614

16 离职公司挽留怎么办？

已经拿到 Offer 了，但现在公司加薪挽留，怎么办？

前一天，参考思路：

- 1) 拿到 Offer，再提离职；
- 2) 如果担心 30 天，公司等不了，面试的时候说正在提离职，预计 15-30 天
- 3) 建议拿多个 Offer，稳妥点；

/

简历 & 面试 21 天打卡，第 3 期 17 天 0615

17 在职面试请假？

面试前：在职去面试，现在公司不给请假，怎么办？

前一天，参考思路：

- 1) 如果确实不想待了，并不是因为薪资原因，果断离职；
- 2) 如果离职是因为薪资不满意，且新公司也没有太好；那就留着吧；
- 3) 如果过不了自己的心理关，那就果断离；

/

简历 & 面试 21 天打卡，第 3 期 18 天 0616

18 定位适合自己的岗位

面试前：如何定位自己最适合的岗位（发展路线）？

前一天，参考思路：

- 1) 控制频率，1 月 2-3 次，每次请 1 天，安排两家面试；具体如何精选面试的公司，见前面的作业；
- 2) 最好利用年假和调休价；

/

简历 & 面试 21 天打卡，第 3 期 19 天 0617

19 内推

哪些方式，可以找到被内推的机会？

前一天，参考思路：

- 1) 测试岗位，严格来说，就三条路：测试开发工程师、业务测试工程师、测试管理
- 2) 或者转岗：产品经理、项目经理、或其他
- 3) 把每个路径的知识体系弄清楚，以及分析自己的优势；
- 4) 敲定后，认准，去深耕；

/

简历 & 面试 21 天打卡，第 3 期 20 天 0618

20 入职前准备

已拿到 Offer，入职前，需要准备哪些（工作相关的，入职资料那些按入职邮件即可，不在这条作业的范围）？

前一天，参考思路：

- 1) 过往同事
- 2) 同学
- 3) 认识的行业朋友
- 4) 测试圈子（让他人知道你的擅长，多分享、多输出文章，有适合的岗位，就是你了）
- 5) 一些有价值的 微信群 /QQ 群（那些整天闲扯的就算了）

/

简历 & 面试 21 天打卡，第 3 期 21 天 0619

21 入职后与预期不一致怎么办？

入职一家公司，工作几天后，发现与自己预期不一致；与面试时聊的相差甚远，怎么办？

前一天，参考思路：

- 1) 入职前，恶补下新公司需要用到的工具 和 技能；
- 2) 熟悉公司的产品；
- 3) 了解这个行业的产品特性（如果是跨行业）；
- 4) 梳理下自己过往了解的知识；
- 5) 好好放松，好好休息；

补充，

最后一天，参考思路：

如果实在不想待，可直接自离；去其他公司入职；

恭喜各位，坚持了 21 天，把简历&面试需要考虑的多数问题都思考了一遍；提前思考，避免面试时措手不及；

建议反复练习 3 次以上，把这些问题理解，深入骨髓。

IDO 老徐

2022.05.02，最早写于 2020 年。

011 · 测试时间评估：三分之一估算法

关于测试时间评估，经常收到测试从业者（测试工程师，测试 Leader，测试总监）的提问，他们没有特别好的思路，期望老徐给一些参考思路。

嗯，@IDO 老徐 确实经常跟一些测试从业者沟通交流 & 了解他们的执行现状。

这篇文章，聊聊 理论 & 现实 的差异化，以及老徐对于时间评估的做法 & 观点。

关于时间评估，有很多非常专业的评估方法 & 评估公式（具体有哪些，不阐述，可以搜索引擎找找）。

实际执行的情况是：没有那么多时间给你去估算；也许，从接到项目到上线，总共只有两天时间（甚至当天接到需求、1 小时后需要你给出估算的测试时间），你期望用 1 天时间去估算测试需要多长时间，明显不现实。而且，不同的项目、不同的团队、不同的质量要求、不同需求的紧急度，需要的测试时间，完全不同。

在需求都不完全清楚的情况下，怎么估算测试时间呢？

老徐给一个简单粗暴的评估方法：「三分之一测试时间估算法」。

具体怎么做呢？

根据开发评估的整体时间（总工时），除以 3，得到测试总工时。再结合经验，适当加减 20% 时间即可。

如果需要把每个模块的时间，细分呢？还是保持如上的原则：总时间不变，等比拆分，得到每个模块的时间。

注：如上的时间，并非一成不变，在企业落地实践中，自己灵活运用。

补充，

1. 测试估算的时间，只需考虑测试的执行时间。如果中途，由于开发延期提测（顺延即可），或者开发修改 Bug 时间过长（20% 缓冲），等待新版本测试（站立会报风险）。在时间评估的时候，需考虑这个时间，把此块时间加上（或者，发版时间，顺延）。

2. 自己估算的时间，如果中途执行过程中，发现时间不够，自己加班，想办法消化（即：自己估算的时间，含着泪也要接受），下次估算时间就有经验了。另外，版本结束后，吸取经验，总结下，是哪个点消耗时间过多，是否可加速（自己总结的经验，终身受益）。

如果确实不可加速，说明整个团队的效能，是低于「三分之一」评估大法的，下次估算，在之前评估时间的基础上，再加上 20% 的时间。

3. 项目过程中，不接受临时新增需求的测试，如果有临时需求，需要增加对应的测试时间（这块，理论上是这样，实际情况是，很多同学，经常被强塞任务，时间却没有增加）。

4. 现实情况是，估算好的测试时间，已确定的上线时间，往往由于一些外部因素，提前上线，这个时候，测试同学，需列出已知风险，并做好本职工作，避免背锅。

end，这篇文章就到这。
有啥疑问，评论区交流。

IDO 老徐

2022.05.03 进行了一些调整，最早写于 2019 年。

012 · 你缺的不是技术（面试了 N 家公司，都是「技术面，通过了」，但在 HR 面，过不了）

有同学咨询老徐，“为什么面试了 N 家公司，都是「技术面，通过了」，但在 HR 面，过不了？（或者 HR 面也过了，在最后的总监/老板 终面，过不了）”

嗯，是个好问题。

这几年，带的团队成员，接触的测试工程师读者，确实有很多这种情况。

一、问题在哪？

@IDO 老徐，也面试也接触过很多这类案例，典型的几个问题：

1、没耐心

面试的一些问题回答，给人一种不耐烦的感觉（也许不是本意，可能就是这样的）。

2、不踏实

对自己感兴趣的事和工作，可以去做；但不感兴趣的，或者没技术含量的，没成长的事，不想做，条件反射式的拒绝；

3、知识面窄

对自己专业内，或者工作中涉及的知识，了解；但，工作没直接接触的，完全不了解，也没兴趣去了解；

4、没目标

- 1) 没确定自己到底找什么工作，找到什么就做什么；
- 2) 没明确自己后续的成长方向；

5、表达

- 1) 表达不够清晰，你以为自己懂了，以为自己讲清楚了，但听者完全不知道你在说啥；
- 2) 自己做过的事，说不清，也说不出产出的价值；

6、期望薪资不匹配

- 1) 期望薪资，与市场不匹配；
- 2) 没去了解市场行情，虚高；

二、优势是什么？

- 1、对技术擅长，有专项优势；

- 2、能干活，能解决技术难题；
- 3、能独当一面，甚至可以赋能团队其他成员。

三、建议

1、走内推，
这种情况，自己去投简历，没有竞争力，陌生面试的通过概率太低。通过内推，解决信任度的问题。只需要正常发挥，展现你的技术面，即可。

2、实在没内推，必须得自己陌生面试的情况下；
参考第一项的 6 点问题，逐一调整；

嗯，加油。
你缺的不是技术，而是「软技能」。

软技能，需要时间来积累、改善、调整。从此刻开始，多看看 @IDO 老徐 的文章，也许能解决你的问题。
这里，专注「软件测试从业者综合能力提升」，不仅仅是技术，更多是解决各类问题的思路。

注：这篇文章，写给新入职职场找工作、或者被迫离职至今还在找工作的你、或者想跳槽换份满意工作的你（任何时候都可尝试去投简历、面试，看看自己值多少钱，是否有公司要；找不到，说明你不够优秀；）

IDO 老徐
2022.05.16

013 · 测试开发工程师误区，你在做无价值的努力

某同学，运气非常不错，有内部转岗的机会。从业务测试工程师，转测开（测试开发工程师）。

领导给他安排了任务，让做一下后续的工作规划。他花了几天时间，画了一堆流程性的内容（一套测试工具，解决系列流程性问题），看着非常完善，没啥毛病。

但，其实没啥意义，
大家在群里讨论的很火热，给了各种建议。我（IDO 老徐）在群里回了一句：太复杂，难落地，推进不下去。

我更建议的是：

- 1、这个流程，可以画，作为长期规划
- 2、当下，去跟业务测试工程师聊，看看他们的痛点是什么，比如，
 - 1) 造数据的脚本，
 - 2) 接口自动化，
 - 3) 回归测试，
 - 4) 高效生成测试用例，
 - 5) 自动生成测试报告，
 - 6) 每周统计质量数据，等等。

这些东西，更有价值。

-- 来自 百人计划 7 期 群里的讨论。

延伸，
很多公司，太多案例，测试开发工程师花了大量的时间，开发了一款牛逼的测试工具（测试平台），最后完全推不下去（完全没有业务测试团队去用），他自己在那苦恼。

但，是否想过一件事：

- 1、你这工具，开发前，去跟业务测试工程师，收集需求了否？
- 2、去观察他们日常工作的痛点了否？
- 3、这工具，如果你自己是业务测试，会用否？

如果你的回答都是，那么找一个团队，先试点。
结果好了，有案例、有数据了，再逐步推开。

IDO 老徐
2022.05.17

014 · 造测试数据的两种方式，三种思路

在测试群及 日常的交流中，如何「造测试数据」，一直都是痛点。解决思路各有差异。

造数据的思路：

- 1、直接去数据库，SQL 造；
 - 2、直接跑接口，**模拟真实用户操作，重复执行**（更推荐这种）
- PS：执行 DB，很容易产生数据不完整性，漏了某些表或某些字段，以及状态的不一致（理论上，不建议做 DB 的 insert 操作）。

注：如果想玩第一种，但 SQL 又不熟的，去看看过往文章，一篇搞定：[软件测试从业者必备 SQL 合集 V1.0 \(21 天搞定\)](#)

这篇，主要聊「**通过跑接口，造数据**」。

其实，很简单，就是接口自动化，**按需跑**一下。

如果你连**接口自动化**是什么，都不知道，建议去 玩一玩 Postman、Jmeter 等工具，Fiddler 也可（21 天打卡，也有这些，可去练习）。

如果能自己写，直接 python 搞定。

如果**想更进一步**，可以封装到 公司自主开发的测试平台，搞一套 UI，按具体业务场景，选择，去一键造数据。

是的，@IDO 老徐 就是这样搞的，非常高效。不仅 整个测试部门在用，相关的开发，日常调试问题，也用我们的这套造数据平台。

很高效。

总结：

- 1、用 Jmeter 串联几个接口，跑数据。
- 2、用 Python 自己写。
- 3、封装公司内部的接口平台，提供 UI 造数据入口，开放给 业务测试工程师 和相关研发团队使用，提升效率。

end

IDO 老徐

2022.05.20，最早写于 2022 年 1 月。

007 · 软件测试工程师转正考核

这一篇，关于「测试工程师转正考核的 方式、题目、方向」。

这个主题，是某位团队 Leader，需要对部门转正的人员出一张转正试卷（这里，不讨论是「转正答辩」好，还是「转正考试」好）。

针对此问题，阐述一些老徐的观点（不一定正确，IDO 老徐也没出过转正试卷这事，欢迎大家补充）。

首先，

一件事，开始之前，老徐喜欢先弄清楚，做这件事的目的是什么，想达到什么目标。

很显然，此问题，考核的目标是：校验，试用的员工，是否达到了公司对于测试从业者能力的要求、业务的要求、基本沟通的要求 等 。

针对，一个测试从业者，必须具备的 「硬技能 + 软技能」需求，以及你对这个岗位的特殊要求 。

试卷，可以从以下几个维度出：

1. 这个岗位，必须要具备的技能 & 工具，弄几个题 。
2. 常用的 Linux 命令/脚本知识，弄几个题 。
3. 产品业务流，弄几个题 。
4. Bug 级别定义，弄几个题 。
5. 发布流程，弄几个题 。
6. 以及，你对团队成员的特殊期望，弄几个题 。
7. 公司的组织架构，也可以弄一个题（员工，还是得了解的）
8. 最后，加几个开放性题目，如：1) 你觉得团队存在的问题是什么？2) 你有哪些改进建议？3) 试用期，你的产出有哪些？

基本上，就可以比较全面的考核一个同学了 。

希望，如上，对你有点参考作用 。

IDO 老徐

2022.05.20 ，最早写于 2018 年，略有调整 。

这份文档，建议收藏 。 很多内容值得反复看，反复研究，反复执行 。

看完这 25 问 + 15 篇精选文，应该可以判断《电子书·软件测试综合能力提升》对你是否有价值 。 如有价值，欢迎订阅（如下，10 元券）

