

聊聊软测人员知识的深与广，解读老鸟成长经历

写在最前面

当提交题目后，我有些后悔了。这个论题太大了，涉及面太广了，不好聊。对于不算老鸟的我，只能是打肿脸当一次胖子。但我觉得这就是我要表述的，希望后来的”菜鸟们”，能有一点点收获我就心满意足了；更希望老鸟们多提意见与建议。

知识的广度与深度体现在哪里

不同人员在同一功能上测试的差异

一个网站的登录页面，是的仅仅是登录页面，你关注什么？

给自己点时间思考,不往后面看，试试自己能想到些什么？



.....给自己点思考时间.....

.....Thinking.....

你思考过不考虑登录和登录后的功能，你关注什么吗？

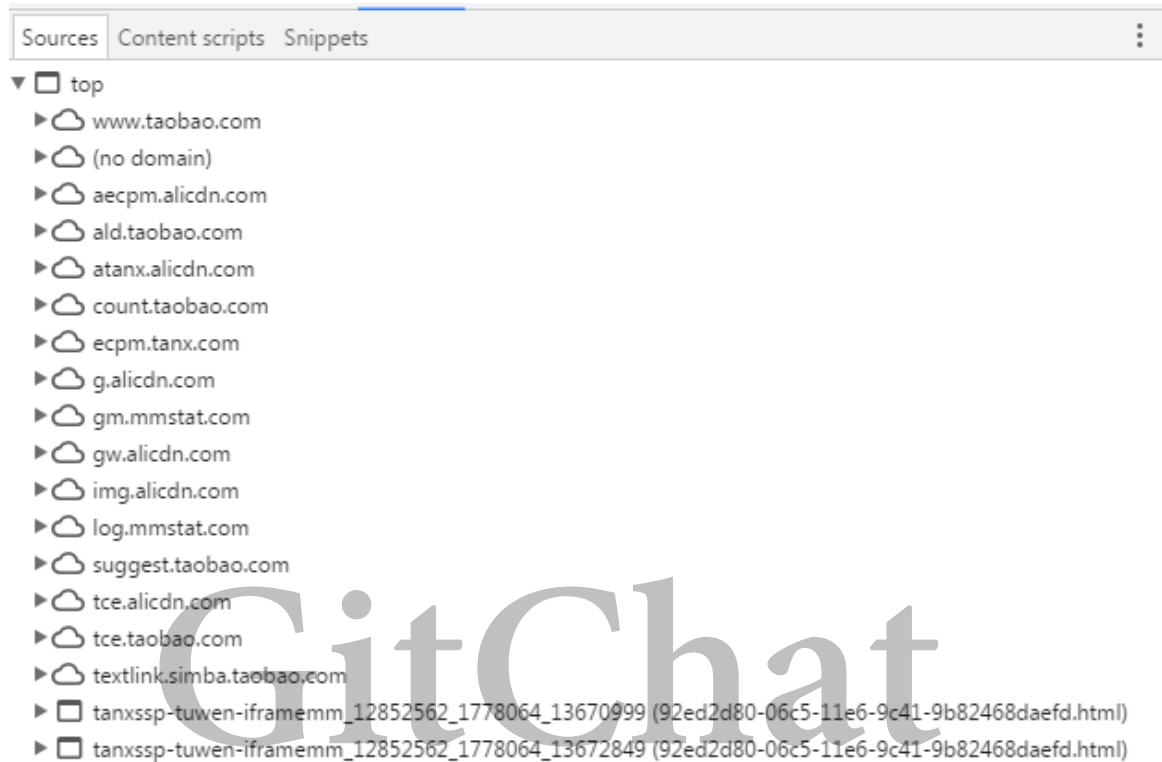
一起来看看主流的思考可能有：

- 关注功能

- 可见的是否展示
- 需要的是否提供

- 还关注上下游

是否从最近的CDN取得资源



通常可以发现网站同步加载，非CDN加载等问题

- 还关注周边

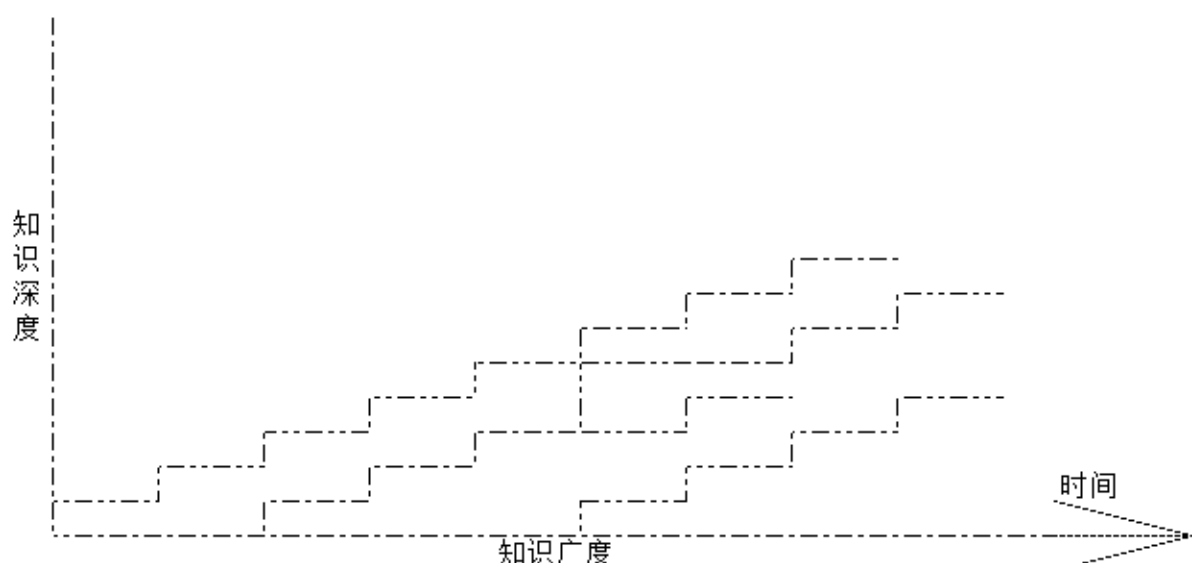
是否进行图片合并，减少请求次数，是否进行图片压缩，减少网络压力

Name	Stat...	Type	Initi...	Size	Time	Timeline
bg1.jpg	200	jpeg	(ind...	179 KB	148 ms	
section4.jpg	200	jpeg	(ind...	131 KB	83 ms	
section4_main.jpg	200	jpeg	(ind...	90.4 KB	58 ms	83 ms
datadt.png	200	png	(ind...	83.5 KB	182 ms	
line.png	200	png	(ind...	72.7 KB	27 ms	
hill.png	200	png	(ind...	63.8 KB	48 ms	
section3.jpg	200	jpeg	(ind...	59.4 KB	46 ms	
section2.jpg	200	jpeg	(ind...	57.6 KB	43 ms	
y_light.png	200	png	(ind...	44.3 KB	34 ms	
xg.png	200	png	(ind...	30.7 KB	32 ms	
logo.jpg	200	jpeg	(ind...	28.9 KB	92 ms	
word2.png	200	png	(ind...	11.4 KB	31 ms	
lozenge.png	200	png	(ind...	7.9 KB	46 ms	
fish1.png	200	png	(ind...	6.6 KB	30 ms	
fish2.png	200	png	(ind...	5.5 KB	42 ms	
Ship.png	200	png	(ind...	5.3 KB	20 ms	
hm.png	200	png	(ind...	5.1 KB	26 ms	
wave2.png	200	png	(ind...	5.1 KB	128 ms	
wave1.png	200	png	(ind...	5.1 KB	20 ms	
icon_map_1.png	200	png	(ind...	4.9 KB	39 ms	
29 / 46 requests 937 KB / 1.4 MB transferred Finish: 772 ms DOMContentLoaded: 686 ms Load: 742 ms						

这是某网站(非淘宝)一次页面请求返回信息，图片资源多达46个，最大179kb。此时测试人员需要给出图片整合、及图片压缩的意见。**同一个功能块，不同人员看到的点不同**，这与知识的广度与深度相关。

知识深度与广度的关系

知识深度、知识广度以及达到某深度与广度所耗费的时间组成知识的三个维度，对于这三个维度谈谈自己的理解。



知识广度有助于提升知识深度，也就是我们常常会说知识有共通之处，但一味提升知识广度无助于解决现实高端问题。什么时候需要需要提升广度，什么时候需要提升深度。

回顾与总结

前段时间有个想学自动化的同学同我聊起，计划明年把自动学了。我问他计划是什么？先学什么再学什么？问完他完全蒙了。

其实无论对于何种知识的学习，一定是先有广度上的认识和了解，才会有深度的学习和理解，只有知识广度得到了积累，才知道学习自动化，要先掌握一定的编程技能，要先有一定的功能测试基础。才能将广度知识的堆叠为自动化的知识。

很多刚进入测试的同学，完全不了解测试应该掌握什么知识、哪些知识。今天的文章就希望给到这部分同学帮助。

让大家认识测试知识的广度，然后去个个积累，然后把广度上的知识进行堆叠和抽象，提升自己在各领域的深度。

在测试自动化上你得先知道有参数化、断言、对象、场景恢复等概念。才可能对文件参数化，数据库参数化有有理解和实践。过程中定期不定期检视自己的学习广度是否达到目标知识，如果达到了，就无需要再拓展可以快熟掌握广度知识的，达到了就可以扩充其中知识的深度。

全局观与拆分

自己学习有很大成本，特别是摸着石头过河。全局观的建立，更建议找个好的指导老师，而不是通过广泛的学习。不需要大师给你特别的细节，只需要大概。如果身边没有大师建议把目标知识的书买来，只看目录，然后把知识的广度扣出来，再把支撑各个知道点的书买来，一样的方法建立自己的全局观。

下一步就设计拆分了，拆分的时候知识点应先易后难，能拆多细就拆多细，最好每个细的知识点能控制到10分钟以内。

有的同学发现自己学一个知识用了两个小时。此时需要回顾，时间都用在哪里了，是本次学习知识的基础知识，是扩展知识，是无关知识？

如果大部分时间都用在基础知识的学习上就需要考虑，改变计划，调整目标。以此来调整自己的目标预期。增强掌握知识的信心。

如果大部分时间用于拓展知识或无关知识，就需要控制好自己的求知欲。管理学习内容和范围，保证学习计划和效果。

另外知识的拆解也要根据自己的学习能力和学习时间来定。掌握以上方法后基本可以遇见何时加深知识，何时拓展知识。

具体步骤

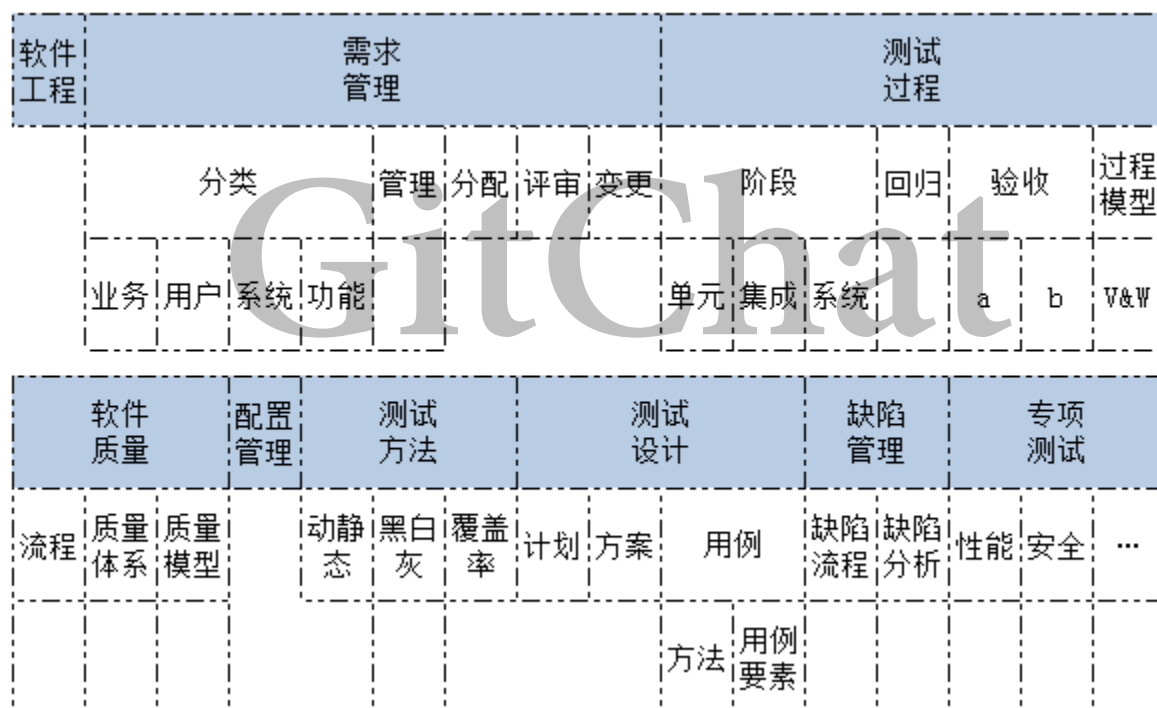
- 定想法（如明年掌握自动化测试知识）。

- 定知识广度（如需要掌握自动化知识要掌握什么？编程，对象库，参数化，场景恢复，思考时间，xapth,持续集成）。
- 定学习计划。
- 拆解学习知识（如持续集成需要拆解：构建工具，自动打包，自动部署，自动测试）。
- 调整为可执行的计划。
- 实施计划。
- 回顾，再调整。

聊聊测试地图

传统开发模式下测试人员需要掌握的知识地图

关键词：规范化、系统化。



地图非常庞大，每个点都需要不断深挖。

如我们要认识过程模型中的V&V模型。V&V是指验证和确认，**验证**是否做了正确的事情，**确认**是否把事情做正确。如果没有测试概念就来谈V&V模型是在做一件有难度的事情。

软件测试是指在一定软硬件网络环境下，遵循合理的测试过程，选择合理的测试工程方法及测试工具平台，使用人工或自动化手段运行或测试某个系统的过程，其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差异。

通过软件测试的概念我们至少认识两点，一软件测试是一个过程；二软件测试进行结果确认。那么V&V模型就是要解决正确的做事，和做的事正确，两件事情。

验证指系统正确地实现特定功能的一系列活动并与这一系列活动的要求相一致。比如传统测试中的需求分析如何验证呢？已经输出了需求规格说明书、完成了需求规格说明书评审、并针对评审进行了修改。我们可以讲对需求分析进行了验证，当然这个标准与公司相关与系统相关，如有的公司要求上线前一定不会有严重BUG，而有的公司则规定问题的ID值不能超过多少。

确认是为满足用户需求的一系列活动，指软件满足了需求要求，解决了相应问题；确认所有过程是否都是在围绕解决用户需求进行开展。

敏捷模式下测试人员需要掌握的测试地图

关键词：连接业务与开发、价值与实用化、效率。



敏捷测试下：测试人员不再只是验证和测试产品，同时支撑业务团队，即支持业务团队与开发团队，做好连接，使团队交付更高质量的产品。敏捷测试下人员的技术能力与一些通识能力有较高的要求。新增的知识有用户故事、实例化的需求、业务价值评估、CI集成、行为驱动开发等；更高要求的知识有仿真、原型、验收测试驱动开发、测试驱动开发。

要了解敏捷测试首先要对敏捷有所了解，比如敏捷4个核心价值：个体和互动高于流程和工具、工作的软件高于详尽的文档、客户合作高于合同谈判、响应变化高于遵循计划。

你会发现每个新增知识点都与敏捷4个核心价值相关，比如CI集成就希望能使用快速的验证来产生一些积极的变化，让团队作出响应。又比如我们希望以用户故事和实例化需求，来提高与合作，而不是大家事先给对方多些条款来束缚生产力。

综合知识

研发类	架构运维类	通用类	辅助类
编程基础	操作系统	文档知识	美工
脚本知识	网络知识	行业知识	外语
数据库	中间件	沟通知识	逻辑知识
大数据		统计知识	
		管理学	

综合类知识与测试知识关联度越来越高。综合类知识对于测试人员解决问题有很大帮助。

特别是在专项测试中体现特别突出，作为一个性能测试人员，不会搭建环境，不会监控系统，不懂得HTTP协议、不知道某中件件如何配置，如何监控是寸步难行的，因此要求人员一定掌握研发类和架构运维类部分知识;而在取得资源得过程中要求人员具备通用类和辅助类知识,如需要网络组同事帮忙提交公司网络及带宽情况时，需要与人沟通；而在确认指定性能场景时，需要有相应的行业知识及与业务专家沟通的能力。

关于圈子

什么样的圈子，决定了你是什么样的人生。

丰富圈子

如何建立自己的测试圈子？

关键词：**自信地走出去，多交流、多分享。**

如果一个项目内你只负责一个模块，那么请你与负责其他模块的同学多交流，那么建立在你们基础之上的，叫项目内测试知识的共性。

如果你项目很熟悉了，请与其他项目多沟通，看看其他项目用了什么新技术，有什么好流程，好方法。

如果说公司其他项目都很熟了，试着联系一下过去同事，建一个社团。空闲时聊聊技术，空闲时做做分享。

完成以上你觉得圈子还不够支撑，那么就试试参加专题沙龙，认识相同行业与不同行业的人员。此时试着找些组织把自己观点讲出来，不断的修正它，不断加入新的元素。

试着找些周边的圈子扩充自己其他领域知识，以此来促进丰富测试域知识。敏捷圈，产品圈，UED圈，营运圈，开发圈，大数据圈等，试着不定期加入，不定期思考，他们的

圈子对于测试带来什么启示与挑战

选出高质量圈子

一直丰富圈子不一定是一件好事，因为你的圈子会消费你的时间。即你的生命，那么如何选出高质量的圈子呢？

总结与检视

每次总结你的收获，如果每次看完文章后，好现对自己一点帮助没有，那么你需要整理一下你的圈子了。

不找最高的，应该找最合适的

有的同学刚入门，就去找了些高阶测试群或者花钱参加一些高级的培训，最后结果大家可想而知。自己搞得晕了，对测试没有了信心。

圈子是用来沟通与交流的

有的同学加了不少群，参加了不少沙龙和主题分享活动。发现自己还是没有多大进步。要注意圈子其它是用来交流分享的，而不是被培训的。

圈子需要精益

所有圈子不能太大，太大了知识点有比较泛了，知识点泛了就会给你带来很多干扰信息。你不得不花5分钟时间读完一个群里的200条消息，可是读完了，发现没有一条是你想要的。但每次又担心放掉了群里最重要的消息，于是每次都会花上5分钟。这样就恶性循环了，所以现在大家都希望看到精品和干货，那怕花钱。

如何成为老鸟

关于路径

我任过教，做过硬件开发，做过软件开发，做了软件测试，带过测试，带过研发，搞过培训，卖过烧烤。渐渐发现发现知识的广度永远会帮忙你叠加为知识的深度。你的理解能力和思维的深度都会得到提高。原来所有知识是相通的，相辅相成的。

在沟通问题遇到难题时，任教的循循善诱会帮助我让同事很快理解问题达成一致；硬件的可视性，让我沟通软件数据流向时更有画面感，更有类别的内容。就连卖烧烤对于我理解用户的需求都是有帮忙的。

所以关于使用什么样的路径可以快速成为老鸟，我是希望所有的毕业生，不要让自己的阅历太单一，一年换个工作是有必要的，当我从某家硬件公司出来时，我觉得我期间我浪费了1年。这种领悟是痛苦的，大家都知道一年能做很多事。那么我给新同学的建议是换换换，多经历广度上的知识，对于未来抽象和堆叠你知识的深度是有帮助的。

另外找一个有挑战的上司，如果你30岁以下，每下都按时下班，有问题有上司帮你担。试问你的成长在哪里。

不要太早考虑职业发展，如果当你还认不清世界，没走过几条弯路，那么什么是捷径，相信你是没有概念的。我要表达的很简单，职业发展是水到自然成的。没有过5年的广泛工作经验，谈什么职业发展哦。

不要太早想挣大钱，30岁前不要想钱，钱是30岁后才考虑的事情。30岁前有积累就够了，其它的该来的时候自然就来了。

关于学习

以教助学，认识学习比学习本身更重要，认识学习的方法比学习本身更重要。相信很多同学已经见过这张图，可以说这张图对我的帮助非常大，也希望给到大家帮助。比如说大家都可以来gitchat分享自己的心得和知识。



以用助学

给自己一个小目标，比如两个月内完成某系统某模块的自动化。

当有了这个目标就有了动力，如果你还担心自己偷懒，试着把自己想做的宣传出去，最好告诉全世界你要做什么，然后让全世界来监督你，帮助你。

关于阅读

找到好的圈子,找到一些高质量的论坛，去那种每日更新7篇的公众号，还不如去看一看每7日更新1篇的干货。免费其实最贵，因为你是拿生命在堆鸡肋中找肉。

专注一个主题

一段时间内专注一个主题，人在一段时间学习后会对知识产生厌恶感。反正我会这样，一段时间对某一个主题很有兴趣，一天看一本书都感觉不够，但一段时间后，在也不想拿起这本书了。

给自己的计划多些调味剂，但所有的计划是缜密的，因为你这一段时间的计划都是为帮助你达成最终的一个目的而定的。

找一个导师

找一个导师有什么好处，一个好的导师能给你系统的理论知识，指导你的广度；另外一个好的导师能给你学习安排，帮忙解决深度与广度冲突的问题，至少能给到一些可行性的建议。

找一个圈子

给你监督，给你交流。圈子的好处在于，大家沟通是处于同一水平，面临着同一难题，有沟通的基础。而大家都希望彼此对彼此进行监督，帮助自己进度

关于书籍

Ron Patton的《软件测试》，最好是原文的，慢慢看就好，多看几次；还有《人月神话》、《测试寻宝图》、《为自己工作》。

关于技术

跟上新技术

对于新技术个人认为红利期在三年左右，也就是说对于新技术测试人员掌握后，可以到两年不用为饭碗发愁，但如果不更新技术薪资可能很快就被拉了后腿。做法很简单，每天花15分钟浏览论坛和公众号，有新技术就多花点时间了解和学习。

学习可用的就好

新技术每天都在更新，学习不完，找到基础的，找到未来3、5年用得上的学习就好，切忌贪多。

关于成长，而不失方向

不忘初心

如果你认定测试方法，希望你能坚持，因为测试路上能做的东西还有很多，至少我们此身是奋斗不完的。常常会看到学习一会说我要学习QTP、一会又找搞Selenium，感觉太累了就换安全测试，最后还是在功能测试。有的同学看到测试工资低了，就在想转研发吧，但转研发不久，发现研发太难了，又希望回到测试。因此希望大家记得不忘初心，才能方向不变。

自信

自信的人没有太多的必要去考虑别人的想法，自己觉得合适合理就去做，大胆的去尝试，如果我们总是考虑别人的想法，那么工作就会有谁多羁绊,而影响自己的计划。

简单

特别是技术方向，大家常常会听到大神们最爱说一句话，我就是喜欢，我就是感兴趣，其实做一件事，很简单的想法最好。

本文首发于GitChat，未经授权不得转载，转载需与GitChat联系。

GitChat