

32 | 软件测试：什么样的公司需要专职测试？

2019-05-14 宝玉

软件工程之美

[进入课程 >](#)



讲述：宝玉

时长 14:15 大小 13.06M



你好，我是宝玉，我今天想与你讨论一下，什么样的公司需要专职测试这个问题。

若干年前，网络上对于软件开发是否需要专职测试有过一次讨论，代表文章有：左耳朵耗子老师写的《[我们需要专职的 QA 吗？](#)》，然后邹欣老师对此回复《[测试 QA 的角色和分工](#)》。

从这些年业界发展趋势来看，看起来很多公司都不需要专职测试了，只需要开发兼任测试工作就可以了。比如，Facebook 号称自己没有专职测试工程师，Google 和 Amazon 虽然有专职的测试工程师，但都是开发人员对质量负责，开发人员写大量的自动化测试代码。但这样真的可行吗？

在回答这个问题之前，我们还是先来看看，软件测试的主要工作是什么？只有搞清楚软件测试的工作，才能搞清楚这部分工作是否可以由开发来替代，是否需要专职测试。

软件测试的主要工作是什么？

在前面开发篇内容的学习中，我们对开发的工作已经比较了解了：在需求确定后，开发人员开始针对需求进行架构设计，然后编码，最后对发现的 Bug 进行修复，保障线上稳定运行。

而软件测试也类似，也是从需求开始，在需求确定后要去对需求进行分析，然后做测试设计。

如果说架构设计是对业务需求在技术方面的抽象，那么测试设计更像是对业务需求的具象，把业务需求分解成一个个具体的用户操作步骤，也就是测试用例。然后在开发完成后，按照设计好的测试用例进行逐一的测试验证，将发现的 Bug 报告给开发人员，并跟踪 Bug 的修复。

如果对软件测试的工作简单总结一下，就是发现 Bug，报告 Bug，跟踪 Bug。

软件测试怎么发现 Bug？

这里面最难的就是发现 Bug，尤其是如何尽早、尽可能全面地发现 Bug。

举个例子来说，如果现在你需要开发一个用户登录的功能，你在开发完成后会怎么测试？

一个普通的程序员通常只会简单测试一下以下用例：

输入正确的用户名、密码，能登录；

输入错误的用户名密码，提示错误，不能登录。

而一个有经验的程序员还会测试一下其他情况，例如：

用户名或者密码为空，是否提示错误；

没有注册的用户名和密码，是否会提示错误。

但如果是一个专业的测试人员来测试，是不是只做上面的测试就够了呢？专业测试还会测试哪些内容呢？

对于专业测试人员来说，上面这些肯定是不够的，还需要有以下这些情况的功能性测试：

用户名密码是否大小写敏感；

用户名或密码如果是用特殊字符，会不会导致程序异常；

用户名或密码如果特别长，是不是会有异常；

是不是所有主流浏览器和终端设备都能使用。

这就完了吗？并没有。除了功能性的测试，还需要进行非功能性的测试，也就是像性能、安全性和用户体验等方面的测试。比如以下测试用例：

是否可以通过发送数据包反复登录，暴力破解密码；

会不会有 Sql 注入的风险；

大量用户同时登录，页面会不会崩溃；

用键盘 tab、回车键是否可以操作。

当然还会有其他测试用例，这里就不一一列举。为什么专业测试人员和开发人员的测试用例会差这么多？

因为开发人员的重点，是放在如何实现功能上，就拿上面用户登录的例子，开发人员会想着如何能校验用户输入的用户名密码，并给出相应的提示，对于异常流程和场景会相对考虑较少。

而对于测试人员来说，重点是在检测，也就是会考虑所有可能的用户使用场景，正常的、异常的，甚至各种极端情况，例如大量并发访问、黑客恶意破解，所以他们能想到更多、更全面的测试用例。

测试人员设计测试用例，就是要尽可能做到覆盖所有用户操作的可能，但理论上来说这是不可能的，因为组合是有无限多个的。不过从测试的角度看，没有必要每一个可能都去测试，可以通过一些科学的方法来通过有限的测试用例，保证尽可能多的测试覆盖。

有哪些方法呢？我给你举几个例子：

等价类划分

就是把所有用户可能输入的数据分类，如果一类数据对于发现 Bug 的效果是一样的，那么这类数据就是一个等价类，测试的时候只要从里面任意选取一个值就好了。比如一个输入框要求只能输入 1-100 之间的整数，那么 1 到 100 之间都是等价的，0 和任意负数也是等价的，101 和之上的整数也是等价的。

因为分类是有限的，这样就可以用有限的测试用例实现尽可能多的测试覆盖。

边界值分析

边界值是对等价类的补充，因为输入输出的边界是非常容易出错的一个地方。比如说上面输入框的例子，0，1，100，101 都是边界值，可以设计用例来测试是否有可能出错。

探索性测试

探索性测试就是根据前面的测试结果，通过有效的策略进行测试。

举例来说，如果你玩过 RPG 游戏的话，里面通常会有走迷宫的地图，各种分叉，不同的分叉可能有不用的宝箱，如果你想打开所有的宝箱，那么就可以在遇到分叉的时候，每次优先走右边的分叉，除非右边已经去过了。

那么这里“右边的分叉是不是走过了”，就属于已经测试过的结果，策略就是优先走右边的分叉。关于探索性测试的介绍可以参考这篇文章《[探索性测试揭秘](#)》。

当然除了以上这几个主要的策略，还有很多其他的策略，比如说：

场景设计；

因果图；

错误推测法。

这里我就不一一介绍，推荐阅读《微软的软件测试之道》，这本书上有很多具体的测试方法的详细介绍。

借助这些方法，测试人员就可以对需求功能设计出完整的、有较高覆盖率的测试用例。

所以，有时候测试人员的工作看起来不过是用鼠标点点测试，但他们在拿到需求后，其实花了很多时间和精力分析需求，然后根据需求设计测试用例，准备测试数据。等到开发人员完成软件开发后，就按照设计好的测试用例逐一测试，这样就可以做到及时发现 Bug。

软件测试怎么报告 Bug?

在测试的过程中，如果测试人员发现了 Bug，就会通过 Bug 跟踪系统提交 Bug 给开发人员。Bug 跟踪系统其实跟我们在《[14 | 项目管理工具：一切管理问题，都应思考能否通过工具解决](#)》中介绍的任务跟踪系统是一样的，它可以方便地提交和跟踪 Bug。

测试人员要做的就是创建一个新的 Ticket，在 Ticket 的描述中，详细说明 Bug 是什么，具体的重现步骤，必要的话还要附上截图、日志等辅助信息。这样开发人员在收到 Bug 后就能快速定位问题，按照优先级对 Bug 进行修复。

软件测试怎么跟踪 Bug?

Bug 的跟踪，并不仅仅是要跟踪开发人员什么时候修复了这个 Bug，通常还包括对 Bug 修复的验证。

开发人员修复完一个 Bug 后，测试人员首先会验证这个 Bug 是不是真的被修复了，然后还要对整体功能做一个回归测试，确保不会因为修复 Bug 而引起其他功能出现问题。

回归测试是指修改了旧代码后重新进行测试，以确认修改没有引入新的错误或导致其他代码产生错误。

回归测试这一步很重要，因为通常开发人员在修复完 Bug 后，只会验证其修复的 Bug，而不会验证其他功能是不是会有影响。但实际上，软件项目中经常会出现修复一个 Bug，而导致系统其他功能出现问题的情况。回归测试，则能有效、及时地发现修复 Bug 导致系统不稳定的情况。

什么样的公司需要专职测试?

了解了测试的主要工作内容之后，我们再回过头来看看今天要讨论的问题：什么样的公司需要专职测试。

如果一个公司不需要专职测试，那么意味着专职测试的工作可以被其他工种替代，比如说由开发人员一起完成软件测试的工作。

想象一下，如果你是一名开发工程师，然后你要兼职做测试，那么你需要额外做好哪些工作？

首先，你在拿到需求后，除了做技术上的设计外，还需要做测试上的设计，借助测试方法设计测试用例。

这样做好处很明显，可以在写程序时，让程序更易于测试，设计时会对需求考虑更全面。缺点也显而易见，你不止要学习编程知识、了解业务，还要学习测试方法。也许对你来说可以做到，但是对于绝大多数开发人员来说，这是一个很高的要求。

然后在开发完成后，要对自己写的程序进行测试。这里可能存在一个问题，也就是如果你在程序实现的时候，漏掉了一个逻辑处理，比如说漏了检测 Sql 注入，那么你在测试的时候也不会想到要去测试这部分。

测试自己的程序还要克服一个心理障碍，就是要对自己的程序进行破坏性测试，才可能找到潜在的问题，但去“破坏”自己完美的程序，对大多数开发人员来说也是很难接受的一件事。

如果上面两个问题都能克服，你还得再考虑一个问题：如果项目进度比较吃紧，作为开发人员你会压缩哪部分时间？

正常来讲，测试时间必然要被压缩的，因为你首先得保证代码实现。这可能导致只要项目进度一紧张，测试就被严重压缩了，进而会严重影响质量。

这样看来，完全由开发人员兼职测试，还是很有难度的，不仅对开发人员要求非常高，而且需要开发人员承担所有的开发和测试的压力。

为什么 Facebook 可以做到没有专职测试呢？

首先 Facebook 的工程师水平确实是高于业界平均水平的，有能力同时做好开发和部分测试工作；

其次，Facebook 的产品周期相对宽松，可以有时间完成自动化测试代码；

Facebook 在功能发布之前，先发布新功能到内部环境，几千内部员工先测试，部分充当了测试人员角色；

Facebook 的发布和监控也比较完善，有问题能通过监控及时发现，并且可以随时快速回滚或者发布补丁；

最后就是用户对这类社交产品的 Bug 相对容忍度比较高，想想看如果是波音飞机上的软件能这么做吗？

至于 Google 和 Amazon 这些公司，他们也是类似的情况：

- 大量优秀的工程师，可以同时兼任开发和测试；

- 有大量的自动化测试代码覆盖；

- 强大的发布和监控系统；

- 时间进度比较宽松；

- 用户对 Bug 容忍度较高。

对于不能满足上面条件的公司，有专职的测试是更有利于软件项目开发和质量保障的。

大厂不设专职测试的启示

虽然对于大部分公司来说，要做到完全没有专职测试还不现实，但这些大厂不设专职测试的实践还是有值得借鉴和思考的地方。

首先，用自动化测试代替重复性的手工测试是必然趋势。随着自动化测试技术的成熟，写自动化测试代码的成本逐步降低，而自动化测试，可以极大提高测试效率，尤其是像回归测试这种需要频繁进行的。

其次，测试设计是软件测试人员的核心竞争力。无论是自动化测试还是手工测试，测试用例是核心。无效的测试用例，用任何方法去测试，都不会达到良好的测试目的，只有测试用例

设计好了，真正做到有效高覆盖，测试才是高质量的。

最后，开发人员和测试人员的更多融合是一种双赢。比如说测试人员可以给开发人员提供测试用例作为测试参考，开发人员可以写更多自动化测试代码，这些方式都能有效保障产品质量。

总结

今天我带你一起分析了什么样的公司需要专职测试。同时也学习了软件测试的一些基本知识，简单来说软件测试的工作，就是发现 Bug，报告 Bug，跟踪 Bug。

要能及时发现 Bug，需要针对需求进行分析和测试设计，把需求具象成一个个用户操作步骤的测试用例。通过一些科学的测试方法，像等价类划分、边界值分析、探索性测试，能有效提升测试的覆盖率。

公司是否需要专职测试，还是取决于公司的具体情况，例如是否有大量优秀的工程师可以同时兼任开发和测试，有大量的自动化测试代码覆盖，有强大的发布和监控系统，时间进度宽松，用户对 Bug 容忍度较高。

课后思考

你们公司有专职测试吗？你觉得是否应该保留专职测试？为什么？欢迎在留言区与我分享讨论。

感谢阅读，如果你觉得这篇文章对你有一些启发，也欢迎把它分享给你的朋友。


软件工程之美

重新理解软件工程

宝玉

Groupon 资深工程师
微软最有价值专家



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。

© 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。

上一篇 31 | 软件测试要为产品质量负责吗？

下一篇 33 | 测试工具：为什么不应该通过QQ/微信/邮件报Bug？

精选留言 (13)

写留言



砍你一刀

2019-05-17

2

老师您好，能发不分享一个比较好的测试用例模板，谢谢
展开

作者回复：我建议你试试testrail，它的测试用例模板非常专业。

对于测试用例，几个关键的字段是：

标题、描述、优先级、分类

测试类型：功能测试、性能测试、回归测试、冒烟测试。。。

自动化状态：没有自动化、只能手动测试、只能自动化、集成CI。。。

先决条件：这个用例需要满足好什么条件
测试步骤：写清楚一步步的执行步骤
期望结果：操作完成后结果应该是什么样的

等等



邢爱明

2019-05-15

👍 2

谁来做测试工作，这也是我一直比较疑惑的地方，和大家探讨一下。
我现在是甲方，主要做的是企业管理软件，业务逻辑和流程控制都比较复杂，部分系统是需要一些领域的专业知识。
软件开发的时候，基本采用的是瀑布模式。首先由专门的人员做需求澄清，分析和设计，一般我们称为业务分析师，他们这些人会输出软件的需求规格说明书，包括软件原型、...
展开 ∨

作者回复: 我的观点是这种情况下需要专职测试的，业务分析师的重点应该是把需求文档写清楚。

业务复杂不能成为一个借口，想想看开发人员是怎么理解需求的，难道也是业务分析师代劳？肯定也是由业务分析师写成需求文档，然后开发人员基于文档开发，当然中间少不了很多确认环节。

测试也是类似，应该专业的人来做比较好，可以有更好的测试覆盖。一开始肯定是难一点，但是一段时间业务熟悉后，会极大提升整个团队的测试效率，而你也不需要再为这个问题纠结了。



拉欧

2019-05-14

👍 2

对于中国的开发，最大的瓶颈一是时间二是心理，需求都做不过来的时候很难考虑测试的完整性；破坏性测试本身是反程序员思维的，要突破这层心理也要专门的训练才可以
展开 ∨

作者回复: 是的，时间是个很重要因素，大部分项目都是时间很紧很紧的。如果时间紧，首先被砍的就是自动化测试。



纯洁的憎恶

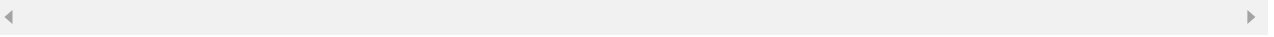
2019-05-14

👍 2

我们公司软件开发采用外包团队，IBM是我们最主要的开发商。但是，开发团队从leader到顾问，再到开发工程师几乎都是IBM在市场上招的临时人员，没有IBM正式员工。给我的感觉是这些人工作非常认真负责、执行力强，但是业务能力确实比较普通。比如他们不用软件工程管理工具，更不要提自动化测试、持续集成和敏捷开发了。还有没有专职的测试人员，业务顾问和开发人员一起测试，回滚也是纯手工，耗时耗力经常加班。我觉得IBM...
展开 ∨

作者回复: 你说的这种情况我有知道，不止IBM，还有微软也这样，微软有一个部门叫MCS，顾问咨询部，他们和一些外包公司合作，自己出几个顾问带队，其他都是外包公司的员工，质量层次不齐。

核心因素确实是成本。



ikel

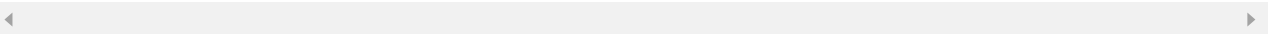
2019-05-15

👍 1

我们公司有专门的测试。就是流动性比较大，新人老是向开发咨询问题，测试内部缺失有力的培训。

作者回复: 测试离职率高是什么原因造成的呢？工资低？工作量大？和开发不和？

我觉得还是得花点时间考虑清楚这个问题，然后让离职率降低下来，让测试团队稳定下来，人员稳定了，很多问题就迎刃而解了。



纯洁的憎恶

2019-05-15

👍 1

看来是否需要专职测试人员，在一定程度上需要视具体业务情境而定。不同的情境会有不同的异常情况和极端情况，需要有针对性的设计出完备的测试用例。而且在bug修复后，也要保证修复本身没有bug。所以测试也是一个系统性的工作，如果取消专职测试人员，不仅对开发业务水平要求更高，还需要项目自身的不确定性低一些。感觉有测试思维的开发人员，更有可能写出健壮的代码。

展开 ∨

作者回复: 👍赞，非常好的总结分享。





谭鹏

2019-05-15

1

移动端的测试 都是靠手工操作 而且是程序员 自测 先把测试用例写出来 给测试。测试看都不看 直接 动手测试

作者回复: 这样的话, 是不是程序员直接按照测试用例写自动化测试更好?



Charles

2019-05-15

1

我经历过一个项目刚开始没有专职测试, 研发部门和其他业务部门一起做测试, 研发部门找到的问题更多的是老师文章中讲的程序员正向思维下的功能bug, 而其他业务部门更像是用户, 主要走的是正常的使用流程, 根本没办法覆盖到特殊的边界类测试, 后来招了专职测试才体会到什么叫专业的测试设计, 虽然可能不是最好的, 但是也比非专职的要好太多了, 线上问题明显减少, 所以我认为专职测试必须的...

展开

作者回复: 这个问题我就真知道了。

有需求就有市场, 只要外包或技术服务这个需求在, 肯定是有市场的, 但竞争多的话, 利润空间已经不会太高了。作出创业初期积累, 倒是比直接做产品靠谱点, 就是得想好未来怎么走。



许童童

2019-05-14

1

如果测试的工资能够给到开发的话, 我觉得作为开发, 就不需要专职的测试人员了。

作者回复: 我觉得这也不完全是钱的事情, 毕竟还需要投入很多时间在上面, 对于开发人员来说, 主要时间都用来实现功能和修复Bug了。

另外要做好测试, 和写好代码还是有些不一样的地方, 尤其是要破坏性的去测试自己写的程序, 总还是要投入一些经历去学习这些专业的测试知识。



kirogiyi

2019-05-14

1

个人认为专职测试是必须的，但他们的主要职责不是测试软件Bug的存在，而是提供系统化的测试解决方案，指导构建完善的自动化测试系统，去引导和培训开发人员完成一系列的测试工作，去监督软件测试的过程和结果，随时提出建设性的意见和建议。最终目标是实现：谁污染谁治理的策略，也就是开发人员要对自己的产出负责，不对测试人员产生心理上的依赖，最大化的从源头上解决问题，以减少问题的传递链路。...

展开

作者回复: 这确实是常见的现象，核心还是多一起合作多相互了解吧，让开发人员看到测试的核心价值，就是对测试方案的设计。

我对测试人员敬佩的地方不在于他们会写自动化测试，毕竟这个我写起来还更好，而是他们总能从我没想到的角度测试出来Bug，从而帮助我提升程序质量。

可以安排一些开发人员和测试人员一起合作的事情，比如测试人员提供测试方案测试用例，开发人员按照测试用例去实现自动化测试，让开发人员明白做好测试其实不是他们想的那么容易的事。

◀ ▶



beiler

2019-06-01

1

老师，最近公司需求，发现手动测试接口效率非常低，我计划做一套自动化测试框架，我之前试用了jmeter，发现写代码太费事，我计划用python写，这样我盯住接口协议即可，不知道老师你们用的接口协议管理软件是什么？是rap2还是swagger，这俩哪个更易于导出json接口，进行自动化测试

作者回复: 抱歉你问的这两个我都没用过。

我建议你可以写两个简单原型分别测试以下，看看哪个更适合你。

不管怎么样，自动化测试的大方向是没有问题的！

◀ ▶



和

2019-05-22

1

我的单位，当然不是软件公司喽，但是企业内部也会开发一些业务类软件。可是从需求分析到设计，开发，测试，部署，运维，都是我一个人的工作。请问老师，这样的情况，您

觉得如何工作，效果会更好一些呢

展开 ▾

作者回复:

对于这个问题，我觉得你可以自己先分析一下，你觉得目前哪些地方做的好，哪些地方可以有改进？

可以从几个方面分析，比如：

工具：

有没有用源代码管理工具？

有没有用持续集成跑单元测试？

有没有用Bug/任务跟踪系统？

开发流程：

多长时间一个项目周期，一个功能从开发到上线要多长时间？质量如何？

有没有做需求分析和确认？会不会做出来东西不是业务部门想要的？

有没有开发前先做简单的技术设计？

有没有写一些基础的测试用例？

有没有开发后自己测试和找业务部门帮忙测试？

线上的故障有没有一个合适的流程去处理？

有没有写一些基本的文档，如果来一个新人了能否接手？

技术实践：

有没有写自动化测试？

有没有用好的技术框架或开源组件？

有没有自动化部署？

分析出问题，知道哪些地方做的不够好以后，就可以有一个改进的计划，看如何将改进方案落实下来。

还有就是一个人开发，缺少向其他人合作和学习的机会，可以有意识的创造更多这样的机会，比如内部多和其他部门合作。外面可以参与一些开源项目。

◀ ▶



hua168

2019-05-20



老师，现在不是流行，测试驱动开发吗？不是先写测试代码再写实现代码的吗？
那还完写完再让专门的测试去测？

展开 ∨

作者回复: 测试驱动是一种很好的开发实践，但普及率也不算很高。

可以看到自动化测试那一篇，测试驱动写的是单元测试，并不能保证不出Bug，只是说能有效提升代码质量。

还有就是开发人员测试自己代码，很容易遗漏编码时就没考虑到的逻辑。

