

输入姓名

大家根据姓名，查询自己的调查问卷，然后在浏览器上选择打印，可以打印到pdf，自己留存pdf。
截止时间：9月25日。

查询条件： 您的姓名：马鑫典

序号：47
所用时间：826
来源渠道：手机提交(微信)

填写时间：2017/9/11 18:43:37
来源IP：117.136.54.123 (天津-天津)

您的姓名：[填空题] *	马鑫典
您的学号：[填空题] *	2117216043
1.保持高标准，不要受制于破窗理论(broken windows theory)[i]。当你看到不靠谱的设计、糟糕的代码、过时的文档和测试用例的时候，不要想“既然别人的代码已经这样了，我的代码也可以随便一点啦。” [单选题] *	我就是这样随便过来的；
2. 主动解决问题。当看到不靠谱的设计，糟糕的代码的时候，不要想“可能别人会来管这个事情”，或者“我下个月发一个邮件让大家讨论一下”。要主动地把问题给解决了[iii]。 [单选题] *	不懂啥是靠谱的设计；
3. 经常给自己充电，身体训练是运动员生活的一部分，学习是软件工程师职业的伴侣。每半年就要了解和学习一些新的相关技术。通过定期分享(面对面的分享，写技术博客等)来确保自己真正掌握了新技术。 [单选题] *	看了就忘；
4. DRY (Don't Repeat Yourself)——别重复。在一个系统中，每一个知识点都应该有一个无异议的、正规的表现形式。 [单选题] *	这要讲场合。
5. 消除不相关模块之间的影响，在设计模块的时候，要让它们目标明确并单一，能独立存在，没有不明确的外部依赖。 [单选题] *	想做，但是不知道怎么衡量效果。
6. 通过快速原型来学习，快速原型的目的是学习，它的价值不在于代码，而在于你通过快速原型学到了什么。 [单选题]	从来没听说过；
7. 设计要接近问题领域，在设计的时候，要接近你目标用户的语言和环境。 [单选题] *	按我的想法设计，用户以后会适应的；
8. 估计任务所花费的时间，避免意外。在开始工作的时候，要做出时间和潜在影响的估计，并通告相关人士，避免最后关头意外发生。工作中要告知可能的时间变化，事后要总结。 [单选题] *	大概估一下，不必在意时间
9. 图形界面的工具有它的长处，但是不要忘了命令行工具也可以发挥很高的效率，特别是可以用脚本构建各种组合命令的时候。 [单选题] *	到时候问牛人；
10. 有很多代码编辑器，请把其中一个用得非常熟练。让编辑器可以实现自己的定制，可以用脚本驱动，用起来得心应手。 [单选题] *	没有任何定制。
11. 理解常用的设计模式，并知道择机而用。设计模式不错，更重要的是知道它的目的是什么，什么时候用，什么时候不用。 [单选题] *	每写100行程序，我就尽量用一个模式。

12. 代码版本管理工具是你代码的保障，重要的代码一定要有代 领导要求才用。
码版本管理。 [单选题] *
13. 在debug的时候，不要惊慌，想想导致问题的原因可能在 只会printf；
哪里。一步一步地找到原因。要在实践中运用工具，善于分析
日志(log)，从中找到bug。同时，在自己的代码里面加 log.
[单选题] *
14. 重要的接口要用形式化的“合同”来规定。用文档和断言、 从来没听说过；
自动化测试等工具来保证代码的确按照合同来做事，不多也不少。使用断言 (assertion) 或者其他技术来验证代码中的假设，
你认为不可能发生的事情在现实世界中往会发生。 [单选题] *
15. 只在异常的情况下才使用异常 (Exception), 不加判断地过 抓住所有异常
多使用异常，会降低代码的效率和可维护性。记住不要用异常
来传递正常的信息。 [单选题] *
16. 善始善终。如果某个函数申请了空间或其他资源，这个函数 一直主动这样做
负责释放这些资源。 [单选题] *
17. 当你的软件有多种技术结合在一起的时候，要采用松耦合的 一直主动这样做
配置模式，而不是要把所有代码都混到一起。 [单选题] *
18. 把常用模块的功能打造成独立的服务，通过良好的界面 拷贝代码过来用也可以
(API) 来调用不同的服务。 [单选题] *
19. 在设计中考虑对并行的支持，这样你的API 设计会比较容易 并行不会出错的；
扩展。 [单选题] *
20. 在设计中把展现模块 (View) 和实体模块 (Model) 分开，这 没搞清楚啥是V，啥是M。
样你的设计会更有灵活性。 [单选题] *
21. 重视算法的效率，在开始写之前就要估计好算法的效率是哪 主动测试程序效率，以验证估算
一个数量级上的(big-O)。 [单选题] *
22. 在实际的运行场景中测试你的算法，不要停留在数学分析层 想用，但不知道工具
面。有时候一个小小的实际因素 (是否支持大小写敏感的排序，
数据是否支持多语言)会导致算法效率的巨大变化。 [单选题] *
23. 经常重构代码，同时注意要解决问题的根源。 [单选题] * 每天应该重构两次。
24. 在开始设计的时候就要考虑如何测试，如果代码出了问 从来没听说过；
题，有log 来辅助debug 么? 尽早测试，经常测试，争取实现
自动化测试，争取每一个构建的版本都能有某些自动测试。 [单
选题] *
25. 代码生成工具可以生成一堆一堆的代码，在正式使用它们之 从来不看那些代码；
前，要确保你能理解它们，并且必要的时候能debug 这些代
码。 [单选题] *
26. 和一个实际的用户一起使用软件，获得第一手反馈。 [单选 想做但是没有机会。
题] *
27. 在自动测试的时候，要有意引地入bug，来保证自动测试的 一直主动这样做
确能捕获这些错误。 [单选题] *
28. 如果测试没有做完，那么开发也没有做完。 [单选题] * 签入代码，就是做完了；
29. 适当地追求代码覆盖率：每一行的代码都覆盖了，但是程序 覆盖20%
未必正确。要确保程序覆盖了不同的程序状态和各种组合条
件。 [单选题] *
30. 如果团队成员碰到了有一个有普遍意义的bug, 应该建立一个 每个bug都是特殊的；
测试用例抓住以后将会出现的类似的bug。 [单选题] *
31. 测试：多走一步，多考虑一层。如果程序运行了一星期不退 从来没听说过；

出，如果用户的屏幕分辨率再提高一个档次，这个程序会出什么可能的错误？ [\[单选题\]](#) *

32. (带领团队)了解用户的期望值，稍稍超出用户的期望值，让用户有惊喜。 [\[单选题\]](#) *

如果有明确要求，我可以做好。

33. (带领团队) 不要停留在被动地收集需求，要挖掘需求。真正的需求可能被过时的假设、对用户的误解或其他因素所遮挡。 [\[单选题\]](#) *

如果有明确要求，我可以做好。

34. (带领团队)把所有的术语和项目相关的名词、缩写等都放在一个地方。 [\[单选题\]](#) *

一直主动这样做

35. (带领团队)不要依赖于某个人的手动操作，而是要把这些操作都做成有相关权限的人士都能运行的脚本。这样就不会出现因为某人休假而项目被卡住的情况。 [\[单选题\]](#) *

一直主动这样做

36. (带领团队)要让重用变得更容易。一个软件团队要创造一种环境，让大家有轻松的心态来尝试各种想法 (例如，模块的重用，效能的提升，等)。 [\[单选题\]](#) *

一直主动这样做

37. (带领团队)在每一次迭代之后，都要总结经验，让下一次迭代的进度安排更可靠，质量更高。 [\[单选题\]](#) *

不但主动做，还会影响同事一起做好

38. 图形界面的工具有它的长处，但是不要忘了命令行工具也可以发挥很高的效率，特别是可以用脚本构建各种组合命令的时候。 [\[单选题\]](#) *

到时候问牛人；

最拿手的计算机语言之一，代码量是多少？（偏前端，pc、mobile app） [\[填空题\]](#)

PC 1000

最拿手的计算机语言之二，代码量是多少？（偏后端，数据处理，网站后台，机器学习，等） [\[填空题\]](#)

c 1300

（阅读代码的能力，实现，单元测试）你有没有在别人代码的基础上改进，你是怎么读懂别人的代码的，你采取什么办法来保证你的新功能不会影响原来的功能？你在开发中刚碰到最复杂的bug是什么，你是如何解决的？这个bug出现的原因是什么，你在将来应该怎么去避免bug再出现？ [\[填空题\]](#)

没经历过

你如何测试你自己写的代码？
你如何测试别人的代码？
你掌握多少种测试工具和方法？
你写过测试工具吗？
你如何对一个网站进行压力测试和效能测试？
你如何测试一个软件的人机界面？ [\[填空题\]](#)

运行一句，输出一句

你写过的最复杂的代码是什么
你是如何测量和改进它的效能的
用什么工具
如何分析的 [\[填空题\]](#)

没经历过

你做过多少个有实际用户的项目
用户最多多少
你的项目有什么创新的地方 [\[填空题\]](#)

没有

你最感兴趣的领域是什么
这个领域过去10年经历了哪些创新
你分析过这个领域前10名产品吗
请分析一下他们的优劣
你要进入这个领域，应该如何创新 [\[填空题\]](#)

人工智能

你参与过项目管理吗
请描述一下两个当下流行的开发方法在你的项目中的具体应用情况
请问你如何决定项目中各种人物的优先次序

没有

有什么理论来支持你的做法
如果你突然项目不能按时完成，你作为项目领导，有什么办法
[填空题]

你做过架构设计，模块化设计，接口设计吗
请说明一下你如何这样设计，你比较过什么不同的设计方式， 没有
你的设计取得什么结果 [填空题]

你是怎么做代码复审的，你加入我们团队后，能帮我们提高代码质量吗 一句一句来
请具体说说怎么提高 [填空题]

你在各种开发平台（web、linux、pc、mobile、machine learning）都使用过什么样的工具
自己写过什么工具来改进工作效率
给社区贡献过什么工具和代码 没有经历过
github有分享代码过吗
你写的技术博客坚持多久，读者最多的是哪一篇
[填空题]

请描述你在项目中如何说服同伴采用你提出的更好的解决方案
过着你如何听取别人的意见，改进了自己的方案 商量着来，一块做
你如何说服懒惰的同伴加紧工作，实现团队的目标。 [填空题]

你上过什么数学，计算机或其他理论课程
请举出具体的例子，说明你学到的理论知识如何帮助你解决实际问题 数学分析 不知道
[填空题]

本科期间，全年级你专业排名多少 [填空题] 10