**P5**

2021-10-4 10:49:16

【问】

首先我想了解一下你的专业背景，还有你做开发的经历，你大概做了多久的时间来开发？

【答】

就是需要具体到做什么类型的吗？我在学校里可能就写了一些课程上的东西，然后我大四去XX实习，然后就做算法方面开发，我从去年11月份到现在应该10个月吧， 11个月，然后做了算法相关的开发，差不多就是在别人已经有的一个算法框架上加一些其他的内容。

【问】

那你的专业背景是计算机科学，就是自己本身在学校里面也做过相关的工作，然后在实习的过程中也做过相关的工作，对吧？好了解了，嗯，第2个问题是我我们这个研究针对的是中国的开发者，然后我们想对比一下中国的开发者和国外的那些那些程序员，那些开发者，他们有什么不同之处，然后就想了解一下，你觉得程序员这个群体，在中国程序员这个群体，他和其他专业其他职业的人相比有哪些不同之处，有哪些特征？

【答】

没理解这个问题前后逻辑的观点，第一前半部分说的是中国程序和外国程序员之间存在差异，后半句在说中国程序员和中国其他职业存在差异，这两句之间有什么逻辑关系吗？

【问】

没有太强的关联，你就根据你自己的理解来说一下，你作为一个程序员对自己的这个看法。

【答】

太宽泛了，我实在不好描述啊。

【问】

就比如说你觉得自己的这种学习的习惯和其他职业的人相比有什么不同？

【答】

可能较大区别在于自学，大部分，嗯，主要是靠自学。

【问】

然后你觉得你的，比如说工作的方式或者是，以及与人相处的方式这些有什么不同，或者在你见过的程序员里面，他们都有哪些比较特别的地方吗？

【答】

可能非常喜欢work from home的模式，就是根本不去上班，就像我现在的模式。其他可能上下级之间较为扁平，对可能就是没有像，但可能是因为我对其他职业不是非常了解嘛，嗯，在非常多的场景下，就是下级可以直接以委婉的方式拒绝上级的指令，如果它的需求是不合理的话。。

【问】

好那我这个问题就大概问问这些吧，可能你自己也觉得这个问题比较突兀，不太好的回答。然后下一个问题是想了解一下你的这种开发的经历，就是我们想了解你最近，也不一定是最近了，就是你之前学过什么新的技术或者新的工具以及新的软件，就可以举个例子，然后希望你具体地讲一下你的整个学习过程是怎么样的，我们希望你讲的是你从听说这个新技术这个新软件开始到学习它，然后学会了它之后用它来做你的开发，完成一个项目或者是完成一个功能的实现，整个过程是什么样的？

【答】

嗯，我就从开始做深度学习这一块开始了啊，我大概是大三上学期就是离现在两年之前开始接触深度学习，不过那时候我还在上学嘛，也不是很有精力，然后就开始先是看看那个基础，比如说看斯坦福的一些网课，像CS231之类的一些网课来了解一下基础知识，也看了一些就是相关的这个统计学习方法的书籍，然后就要开始写代码了，写代码我发现很难，根本不会写，然后就上网去查呀，就都说有两个框架，一个叫tensorflow，一个叫pytorch，这两个很火，然后先去试tensorflow，然后发现那个静态图写也写不明白，就它有一个子库叫做Keras。 Keras是一个动态库，感觉那个比较好写，基本上就拿那个开始写，然后先从网上抄了一个方法，然后在上面一行一行改，几乎是写一行在百度上查个二三十分钟才能写一行，然后就一行一行改，一行一行看，然后旁边也有高年级的博士学长来指导我，然后就这样就开始入了门，写了第1个很小的项目，可能加起来也就不到1000吧。然后后来就是去商汤那边了，在商汤那边之后也是，我就是在一个大的这个算法框架上，就是他们搭了一套自己的框架，然后我在上面加一个文档版面分析的这个功能。嗯，但是在加之前，我可能先要知道他们那个整个框架在干什么，对吧，这就是他们那做的不太好的一个地方，就是他们的文档几乎是0。整个上万行代码，它的文档就只有一个read me，然后进去以后几乎是崩溃了，发现啥也干不了，只有那个框架，只有两个开发的人知道他是怎么用的，其他人都不知道他是怎么用的。

【问】

你说的文档是指这个项目团队内部的文档？

【答】

对。嗯，然后怎么解决呢？然后他告诉我是这样的，然后告诉我这个框架是抄网上那个，我忘了是哪个开源，是谷歌开源的一个框架，然后让我去看这个谷歌开源的那个文档，但确实写得非常详实，就是这个接口啊，还有文件中，各个函数作用都写的比较全。然后那就先看看谷歌的那个，看了七七八八大部分都看完了，然后少部分也看不懂，因为这个光看文档不看代码还是有点虚，然后就把谷歌那个看完，半懂不懂的时候就硬看他们写的那个框架，然后看了大概一半多的代码，大概明白他要干嘛，就可以往上面加，但时至今日我有很多代码也没看懂，也不知道他在干嘛，就是懵懵懂懂的状态，就是如果在这个框架上做开发的几个人都离职的话，基本上这个框架就没有人能看得懂他了。

【问】

那你能具体的讲一下这个你看到的谷歌的开源文档，它是什么样的吗？你是怎么用它的？大概的结构内容等等。

【答】

嗯，它进去有一个啊，就是类似于read me这样的东西，告诉我这个干嘛，有哪些功能和特性，然后怎么样来使用它。然后进去之后先对于它的各个几个主要组件有一个大概的概述，然后他就，这个更细，就是对于每个文件进行概述，对于每个文件中的每个函数就会有一句话描述下功能，然后对它的接口的各个变量给出功能上的解释，就这样非常详细，一个函数一个函数的讲。

【问】

因为你一开始也没有接触到相关的文档，你说你们项目内部的那个文档就几乎是没有，几乎就是0，然后你看谷歌的文档的时候是从头都看一下吗？大部分都看了对吧？嗯，好，那还有其他的吗？除了看这个谷歌的文档，还有其他文档吗？

【答】

有，我有一次有一个视觉分割的算法，然后商汤已经有这个产品了，然后内部也能用，然后他们内部有一个，忘了是什么平台了，反正就有一个技术文档的这个网页，然后他整个算法作为一个工具，然后在这个网页上就有个描述，就是大概有4页吧，然后第1页就是对他整个做了一个概述，就说一下它主要的功能，第2页就给了他的各个，使用不同参数下的性能的一个表格，第3个就给了一些可能调参的时候他的一些比较重要的调参的那个变量的含义。第四，也忘了是什么，嗯，就是有这种技术文档的时候，然后我使用它就非常方便了，就说我要去调参。但是它这个可能是它的开发文档没有向全公司范围公开，我也看到了他的开发文档，那个文档几乎看了之后就能上手用了，你知道调参非常方便，一下子就把结果做出来了。但是它也有没有覆盖到的地方，比如说input output文件放在哪里，他这些都没有告诉人，需要人自己到代码里面去看，然后对于一些变量描述也没有全，感觉问题挺大的，相当于在迭代开发的时候，就下一版的这个软件，可能还在用上一版的文档。

【问】

你说的这个文档相当于公司内部的文档，还没有对外公开，但是对于公司的开发人员他是公开的，然后你觉得它质量也也有好的地方也有不好的地方是吧？

【答】

对对对。

【问】

大概了解了，嗯，然后下一个问题是想了解一下，嗯，你对这个文档的认知，你觉得文档对于开发工作来说重要吗？极其重要，你能够再详细的展开说一下吗？

【答】

文档是一个为未来而做的工作，而不是为当下而做的工作，显然他要写一个文档可能有时候比开发本身要消费更多的时间啊，但是这个工作是为了未来做铺垫，比如说未来出现的场景，比如说我需要去维护他，过两个月之后我写的代码，我写的什么我自己都忘记了，那这个时候有一个文档可以帮助我自己快速回忆起来，这可以用于一个人的在时间上的差异，用于记忆回忆，也可以用于，比如说某同事离职之后，他的这个文档，嗯，他如果没有留下文档的话，那基本上他做的那个项目在要靠另外一个同事替他想起来这些，每个变量设置有什么作用，整体的架构设计精巧度在哪里，这些都如果要重新从头再来的话，基本上都是要花很多很多精力，而如果他当初在离职之前或者是他在开发的时候就已经留下这样一个文档的话，可以非常大地节省这方面的精力。刚才说的可能是技术问题，我不知道需求文档算不算这个访谈内容。

【问】

需求文档如果对你们的开发工作来说也有用的话，也可以算。

【答】

在开发工作的时候，其实我经常向我的leader要需求文档，可是他们从来不给需求文档，是因为他们自己说不清楚自己的需求，嗯，但是我还是觉得一个好的需求文档是非常有益的，因为好的需求文档，说细一点，可能就是output的内容，是一个文件啊还是只是内存中的变量，那内存中的变量我用什么map的方式，list的方式，还是打成一个tensor data set的方式，就是说他这些都没有规定好，他就告诉我啊，输入应该输入什么，输出应该输入什么，其实这对于开发来说是一种非常大的弊端，一开始我可能做好，然后告诉我那边接口是怎么怎么样，这种靠口头交流，完全没有需求文档来靠谱，而另一方面需求文档也是帮助这个提出需求的，要履行自己的需求，比如说可能就说一下任务嘛，就是就是已拍脑袋一说，这也不严谨，也不一定经得起推敲，但是写成文档的时候可以倒逼那个提需求的人去思考。

【问】

我大概了解了，你说的这个重要程度主要主要是针对开发中间过程以及开发之后的，就是对未来的一个事情，那你觉得他对于你开发之前有用吗？就开发之前你有可能需要选择技术或者框架，那就有不同的选择，然后你会依据文档来进行选择吗？选择之前你会去看文档吗？

【答】

看什么的文档？就各类技术文档？

【问】

对。

【答】

嗯，应该说是在我在写开发文档，就讲道理，还是有一个什么我记不起来了，我记得好像之前在学校的时候是一开始不能先这么写，大概就要先写一下这个框架怎么设计的，然后再去写那个细的开发的那个文档，但是我忘了之前那个叫什么名字，嗯，那您说的那个情况应该是写前面整个技术框架的文档之前需要调研哪些文档？去调研那些文档当然是非常重要的，能让我去比较正确的判断我应该用什么技术来实现。

【问】

大概了解了，好就是下一个问题是你在开发的过程中遇到问题会怎么办？你可以举一些例子，bug啊或者是你不懂的，不知道怎么写代码，不知道怎么做一个功能，这种问题。

【答】

一般都是先自己想办法自己解决吧，也不太好意思老是麻烦别人，然后如果是觉得有框架上的问题，一般问这个框架的开发者他当初设计某个变量或者某种设置是什么含义。

【问】

那你要自己解决的话，你是怎么解决的？你会去看文档吗？

【答】

文档，没有文档，一共就只有一个文档。

【问】

那会是会去百度搜索是吧？搜索别人之前的回答。

【答】

对对对。

【问】

那有遇到过通过文档来解决问题的情况吗？

【答】

之前我说那个视觉分割，之前跟你说过公司内部的那个文档的事情，视觉分割的时候我一直搞不清楚，那个像素是0代表背景还是1代表背景还是255代表背景，然后看半天也没看懂，然后一看，发现文档里说的是255代表背景，那代码实现的时候发现好像是把0当成背景了，然后搞了半天最后也搞不明白，因为他文档老是比那个技术要落后，最后就问了一个用这个框架的同事，他告诉我是其中一个是代表背景的。

【问】

就其实文档也解决不了是吧？

【答】

不不不，它是文档解释错了，因为文档没有跟上技术开发的，版本没跟上。

【问】

那你平时会用到哪些类型的文档？你可以列举一下你用过的文档的类型，技术文档的类型。

【答】

嗯，应该可能有两类吧，第1个就是比如说产品发布的时候就是会有一个页面，然后这个页面上就写了这个新的，就是刚才说那个分割算法，然后他第一页会列举这个特性，如何安装，有哪些功能之类的，然后第2部列举有哪些调参，这种都属于主要从表面上去利于用户的这样一种文档，第2个是开发者，开发者所用的文档都会记录，比如说某个函数有某个作用，然后某个接口是某个含义，然后整个系统构架设计会有哪些部分。主要是这两种文档，一种是面向用户的，第2种是面向开发者的。

【问】

那你一般会通过什么样的方式来找到您需要的技术文档？刚才提到了搜索，除了搜索还有其他的吗？

【答】

向人要，比如说我要买框架文档，那谁在用这个框架我就直接去问他就知道了。

【问】

你说的人指的就是这个公司内部的人是吧？他们是有现成的这种pdf的文档？

【答】

可能是md，也可能是在网页端，他们会在一个网页上持续的更新，然后就让他们分享给我。

【问】

那自己需要的那种，就需要解决问题的那种文档就是通过搜索是吗？

【答】

对，基本上百度搜出什么bug就是什么bug呗。

【问】

那用到的搜索引擎只是百度吗？

【答】

百度、Bing、Google都有，知乎应该也是。

【问】

你提到知乎，你会用那些博客吗？其实在我们的分类里这些技术博客还有程序员他们的社区，这里面的这种问答也算文档。

【答】

当然会用。那我也经常使用这种文档。

【问】

那你可以列举一些吗？国内国外的。

【答】

国内的就像那个csdn，还有博客园还有知乎，这上面有很多就比如说报错吧啦吧啦，然后它会有一系列解决方案，然后还有各种技术教程之类的，然后国外的话就是解bug最多还是stack over flow，主要是报各种bug也能解啊，然后想想还有什么，还有GitHub应该也算一个，GitHub issue中可以提各种各样的bug，然后会有这个GitHub仓库的所有者来回答。嗯，那这个在广义上算，那是Gitlab也算了，跟GitHub几乎是一模一样的一个东西。可能国外也有，但是点网页的时候没仔细看，包括还有像那个像爬虫战之类的，我不知道是不是违法的啊，就是它会有一种，就是他会专门找那种程序员的博客界面，然后他用爬虫把这个给扒过来，然后把它这个爬来的放到自己网站上，然后它爬来了之后，除了那个格式可能有微小区别之外，内容是完全一样的，然后再进网站它就会放一些广告吸引人点进去，然后点进去内容确实也是别人的博客，但是没有尊重别人的版权，但是上面确实也可以看到别人是怎么解决问题的。

【问】

明白了，还有其他的吗？

【答】

可能知乎里面可以分问答和他的专栏，专栏就类似博客，然后问答就是问答，那这样子来说的话，像csdn的评论区也算是一种文档，那会有一些提问和回答的内容应该能算，哔哩哔哩应该也能算，它本身就是也有课之类的，包括我自己也在上面发过，就比如说，如何使用某个编程软件，如何进行debug这样的视频，它本身虽然是视频形式，但是你也算问答，算一种教程。

【问】

那就你看过这么多的博客，还有这种官方文档你应该也看过。那看过这么多的文档，你觉得他们的质量怎么样？我们可以分国内，还有国外的来讨论。

【答】

就国内和国外可能这个分类就太宽泛了。

【问】

你可以就大概的说一下你自己的一个主观的认识。

【答】

我觉得国外开发者对，就是至少对我能看到的文档方面，就是可能我不能否认，就是有些国内开发的他们有文档他不肯放出来，我不能考虑这种偏差，就是他们，我只观察他们放出来的情况，我觉得像谷歌之类的大公司，像Hugging Face这样的大公司，应该还有没有上市的，这样的国外公司，他们对于文档的开源，包括像代码注释，本身也是也是一种文档，他们对这种无论是代码注释这种文档，还是说传统的这种文档都做得远比国内公司好，国内公司基本上文档都是点到即止，蜻蜓点水写一些，可能几万行工程文档就只有几百几千个字。但是国外开发的这个文档，可能他光是字数就能和代码相媲美，就是相比的一个数量级，这样我觉得国内对于文档的重视，包括在工业界和学术界，学术界更是毫无文档可言了，基本就是一个前面的学长开发完，后面一个学弟基本就只能听他是怎么讲的，他自己也不写注释，也不写文档，几乎就是，学术界就是没有这种写文档的习惯，在工业界，就我看来的情况，在做算法的人这里，就只能说在商汤这里，我看到的情况是，文档并没有在开发中起到非常大的作用，而是靠人作为信息传递介质，这不是一种比较健康的环境。相比之下我觉得国内是没有国外做得好，但是我也没有做过国外的工作，我也不知道他们工作中对文档是怎么样的态度，我只能看他们放出来的。

【问】

放出来的文档，你觉得它整体的质量还是比国内的要好。噢，那你可以回想一下，就是举一两个例子，嗯，举一两个文档的例子，然后你觉得比较好的，一个是比较好的，一个是比较差的，比较典型的那种例子，然后说一下它好在哪里，不好在哪里？

【答】

好的就比如说有个公司叫Hugging Face，Hugging Face专门就是把别人训练完的一大堆，就transform这一系列模型他都放在他这个网上，放到他的这个网站上去，然后别人都可以通过它这个网站去下载别人已经训好的这个模型，然后它对于每一个这个信号模型，他的代码，实际上它的代码中就会，他会去重写别人的代码，为的就是让这个代码变得可读，然后它自己会在上面加很多很多注释，然后自己再给出一个文档来，方便这个下载这个代码的人对它进行二次开发，也方便他们进行阅读嘛，嗯，就是相当于他用自己的公司就是去做一件公益的事情，然后来方便别人，就是方便二次开发者，对于原来发布这个模型的人，发布的模型有一个更方便去认识它的途径。

【问】

这个文档相当于是别人写好的文档或者代码，然后他借用过来进行了二次加工，不是他原创的文档？

【答】

对，他自己原创当然是有原创，他有可能是自己公司写的，然后一部分是别人写的，然后共享到他这个公司里面。

【问】

那他原创的那部分质量怎么样？

【答】

非常之高。

【问】

你可以从内容上还有它的结构上来说一下，好在哪。

【答】

内容就是，他有一些所谓的就是自己开发，就这样，就别人发了某某论文，然后有一个仓库里面是他自己的代码，然后那种代码往往是写的很，不能说很烂，它整个发不出来的东西就是很不可读。然后他会看了这个代码，然后看了他的思想，然后自己从头到尾写一遍，然后从头到尾写一遍的时候他会有，就会有这个，比如说python的文件，然后在文件里面会写每一个函数它的作用是什么什么，然后对于它输入的每一个参数都会有一句话去解释它到底是干什么什么用，然后对于程序中所谓他觉得不好理解的地方会加一个注释来写，我为什么要这么这么写，或者他自己没搞懂就说，我也不知道为什么要这么写，但是原作者确实是这么去实现这个方法的。然后它整个发布出来，估计会有一个很大的文档，然后很详细的描述了这个干了什么，我跟原作者有什么区别，我基于他做了什么改进。然后我设计有哪几个模块，就写一个非常非常详细的一个文档来帮助用户来理解，所以经过他二次开发的这个项目反而要比原作者的远更受欢迎。

【问】

就是说它从内容上来讲会更加的详细更加全面，另外一个就是可读性更强，明白，那从这个文档它的组织结构上来说他有什么变化？一个文档，因为它有很多内容，他肯定在这个设计上要有一定的逻辑。

【答】

嗯嗯，可能还是比较传统，就是两级吧，第1级描画层级，第2级描画文件。没有见到什么太创新的地方，就是那种比较传统的那种排列的方式，但是可能他做的比较好的一点就是这个代码间注释，是用注释来替代文档也比较多吧，因为有挺多开源软件也没有对每个文件有很详细说明，因为没有必要，他代码注释中已经对于每一个函数、每一个街口就有很详细的定义，你再写一遍，说实话，就是把那个复制粘贴过去，也没有必要这么写，那种代码中注释来替代文档，我觉得是做的非常的好的。

【问】

你能想起一个比较失败的文档的例子吗？

【答】

我在商汤做的就是很失败的文档，基本就是开发者就是不想写文档，是因为他们这个开发跟传统的开发不太一样，传统开发基本是先有了，别人有需求，然后有需求文档，然后需求完做构架设计，有构架设计的文档，完了之后就是更加细节的开发文档，然后就这样一个流程，就是叫什么瀑布模型，这种就是一级一级去做开发的，但是他们开发就很随意，就随便，就自己就开始码了，然后码完之后再去加文档。其实因为他这个也没有办法加，他写之前他自己对自己需求也是不清晰的，构架设计也是不清晰的，基本就是从别人那复制过来几个文件，然后自己调一调，把它调通了就好，他就没有写文档的这个条件。然后最后给出来的文档也是只有一页，然后就几乎没有参考价值了。然后这个写的时候就带来一个不好的影响，就我在后面做后续开发的人就再也不想去写文档了，因为我的理由就是别人不写，我为什么要写，我不就成冤大头了吗？

【问】

你说的这个还是公司内部的这种文档，然后你能举一个公开的官方的文档，在网页上可以直接访问的这种文档的例子吗？做的差的。

【答】

太多了，比如说在IEEE上发的一些文章上面，它就有些会开源自己的代码，开源就在GitHub里边一个项目，然后这个项目点进去他当然也会有对这个项目的介绍。这个项目介绍，就比如说我看了篇别人写的论文，如果你想查，你直接搜索一下就能查到，就是这些论文里面，应该说它开源的代码里面，可以说他的那个文档的几乎就是很简短，就是一页，然后对于我做开发也没什么帮助，就基本自己去看代码，才能我了解他这个项目，就其实大多数开源的论文的这个文档质量都是很低的，因为都是学术界，大家都不怎么留文档的。

【问】

嗯，然后就是想问一下，你在阅读文档的时候，就是已经找到这个文档，然后你怎么快速的找到自己需要的信息。打开这个文档之后，怎么快速的找到自己需要的信息。

【答】

就CTRL + F。直接用他的文档编辑器查找功能就可以了。

【问】

那除此之外就是会看它这个导航栏啥的吗？

【答】

会，如果它有逻辑结构的话，当然可以通过逻辑结构去查。

【问】

就你最习惯的阅读方式还是CTRL+F搜关键词，直接定位到那个地方。嗯，然后下面一个问题是你写过技术文档吗？你觉得难点在哪里？你可以具体的讲一个例子。

【答】

写过，我想想看哈。难点应该在于没有耐心。

【问】

没有耐心，是一开始就排斥这个事情是吗？

【答】

因为怎么说呢，就是做算法开发，它不是像传统开发就是那样有瀑布模型，就是我很清楚知道我应该要去做什么，然后把它一步步去完成，而这些算法开发它是一个不断的试错，就是大部分代码都是没有用的，之后会被修改掉或者是被删除掉，就是在一个探索中前进的过程，就是探索前进的过程，就注定了他很难是那种文档伴随代码，就是我写一些文档，写一点代码，写一些文档，写一点代码这样并行，可能确实得等到他们都写完了，比如说我都已经调试通了，证明了这个思路还是可行的时候才去写文档，然后这个时候写文档对来自己来说就是没有一种眼前的收益，更多是长远的投资，但这个投资不一定回报的是我，因为我那时候可能已经离职了，就是没有耐心。

【问】

一个是这个出发点就不太，就没有这个动力去写文档。那如果不得不写呢，就真正的写起来的时候，你觉得难在哪？

【答】

难在这是一个很花费精力的事情，就是写文档都要花费写代码那么多精力，因为需要把每一个函数、每一个接口都要非常详细的写清楚，他本身是非常花费精力与时间。很难，我觉得很难，绝对很难。

【问】

就是描述这个就很难。

【答】

对，想要把它写得精确，没有二义性的话就是非常难，而且这个可能是跟学校的培训有关系，在学校里基本没有受过我怎么样去开发一个别人看的懂的文档这种培训，第1次做的。

【问】

了解了，那就最后一个问题啊，就是想让你总结一下什么样的文档才是好的文档，就是你希望在你以后的工作当中，你希望你期待用到什么样的文档，你可以描述一个理想化的状态。

【答】

可能有不同的视角吧，是作为使用者还是？

【问】

你可以罗列几个点。

【答】

作为使用者的话，可能就是希望能看到代码版本和文档版本对齐的这样一种状态，然后写的时候我不仅能看到我该如何，就是它本身这个项目依赖于哪些环境，我该如何下载安装，它的功能概述和它的特性特点有哪些，然后以及他能达到的效果的，表格概述，然后还有就是它的一些可调参数性，以及可调参数范围，是离散的还是连续的，都希望能给出一个范围来。以及有这个项目常产生的问题，他会给出Q&A的一个列表来给出answer。然后还要给一个这个项目目前维护负责人的那个联系方式，让我能够有问题可以直接联系他，就是有一个反馈的途径。开发文档确实有其困难之处，这可能要取决于开发模式吧，如果是像瀑布模型这样的，就是一级一级往下走，当然是有需求文档，构架设计文档，然后到最下面的详细的开发文档，以及有了第1个才有第2个，有了第2个才有第3个，那现在想想这种敏捷开发之类的可能对文档本身就不是有那么大依赖性。对于不同的开发模式，对文档的这个需求或者说期望也是不能混在一起说。

【问】

现在用的比较多的还是敏捷开发。

【答】

对，更多的是乱开发，就乱开发，就给自己挂敏捷开发了。

【问】

那就是这种敏捷开发，他对于文档的依赖程度不是很高。

【答】

不是很高，有的时候就是没有办法构建出一个好的文档来吧，可能就是整个，他整个这个开发流程迭代，就迭代到了一个阶段的结束的时候才能给出，就专门去写一次文档，这样就可以在开发过程中写文档好像不是一件非常容易的事情，这可能是也是因为别人没有探索出这样一个机制来吧，我听说XX那边会有code review这样的机制，就是就是下级的那个代码需要经过上级，我不知道文档会不会像代码一样需要必须要经过上级审阅之后方可并入代码库，在商汤这个工作没有人在做。没有code review机制，就是你加进去就加进去，没有人审核你的代码，然后文档也是加进去就加进去，你不加就不加，也没有人管你，他们会用一些代码分析工具，就所谓的那些，就是自动化检测你的代码，但是这个几乎没用。

【问】

除此之外还有什么你觉得比较好的文档的点，设计的点？

【答】

我觉得好文档应该，本身就是开发者投入很大的精力去做的东西吧，这说到底还是，虽然有些困难，但是说到底还是人的主观意识的问题，就是你愿不愿意留一个好的文档，那有的人就愿意，有的人就不愿意，所以就有好的文档和坏的文档之差。就虽然刚才也提到，就觉得就是写文档会有一些困难，但相比写代码其实是真的不大难，因为写都写出来了，那还能描述不出来不成，那也不不太可能啊，对，就是说到底还是人有没有这个意愿，就是生产时候有没有这个决心去留一个文档。那有决心，我觉得做什么，可能描述不太好，那完备肯定是可以完备的，那没有这个决心就瞎混，反正我到时候离职啊，领导也不管，这样一种心态去做，那就是做不好这个文档，可能代码开发就是做好了，但文档就是留不好，说到底还是开发团队这个leader的意志所决定的。

【问】

那你可以从文档本身，就是写出来的这个文档本身的一些维度来来说一下，它好在哪里。就你觉得他需要做到怎么样的程度才算好的文档。

【答】

那取决于它是什么样的，是给什么样的人看的文档才能回答这个问题吧。

【问】

比如说就是给开发的人看的。

【答】

给开发者看的问答他需要包括，可能需要有一个需求的概述，然后对于整个设计构架，比如说给出构架图，然后给出哪几个主要部件，部件的功能，以及它们是如何耦合的，能给出一个类似于UML图的东西，然后对于具体的代码层面还要给出每一个函数的功能，然后他输入的这个，输入参数，输出参数的类型以及它代表的含义。还有一个比较重要的是能，你比如说在代码层面上他加注释说明在某个部分为什么要这样设计，要不然别人阅读的时候就不知道，对讲清楚原理，甚至很细节，你可能是一行，但是一行就是让看的人不好理解，需要加一些注释来说明为什么这么做。还有可能就是需要有一个专门的人，他一直就在这个，比如说这个文档就是在这个项目上做，然后这个人本身就是一个活的文档，能给我解答一些文档上所没有覆盖到的问题。

【问】

你是说有个联系方式，然后像客服那种东西来给你解答问题，对吧？

【答】

对对对。

【问】

还有其他的吗？那从设计上来说，你有什么期待？交互设计。对，问答的交互，比如说UI设计，放在网页上的文档，它其实也讲究设计的。

【答】

就可能类似个PDF，就打开一个pdf的时候点击那个目录，那不是会有比如第1章第1节嘛，跳转，对对会有这样的跳转结构。我觉得刚刚看那个就挺好用的，所以交互式UI可能就是Q&A吧，就比如说我在GitHub上我提一个issue，然后过几天会有个人来解答啊，我觉得这个也是非常好的，因为文档是再全他也不可能描述，就知道所有问用户会用到的问题，必然还是要有人来作为文档的最后那个托底的方式的。

【问】

那你读过国外的文档，你觉得语言是一个问题吗？读那种英文的问答。

【答】

取决于一个用户的水平，那水平高没问题，水平低就有问题啊。

【问】

你觉得有问题吗？

【答】

我觉得问题不大。大多数都能看得懂，因为他其实技术英语并没有那么复杂。

【问】

你的回答跟前面几位访谈者的回答差不多，就是他们都觉得读这种英文的文档可能问题不是很大。

【答】

对，可能比中文还要清晰一些吧，因为它本身变量名是英文的呀，那中文英文对应可能还挺麻烦。但阅读速度肯定会慢很多的。

【问】

还有就是刚才你提到的这个反馈啊，然后我们有一个问题就是如果那个文档的下面有一个反馈的按钮或者是反馈的框吧，然后你愿意给他反馈吗？如果你发现这个文档有问题，你愿意反馈吗？那如果这个文档写的实在是太差了，你想怎么反馈呢？

【答】

当然了，愿意反馈。可能如果不是在公司内部可能会喷他两句，但是其实一般我也不会这么做，毕竟是怎么说，人家不是在公司内部，人家就没有为我提供服务的义务啊，他本身就是一个对开源社区的一种贡献，可能就是他因为他水平或者经济原因没有贡献好，一般还是，就心里肯定会喷他，但我不一定会发上去喷他，因为这个就不是一个文明的开发者的习惯。

【问】

那你还是愿意评价，愿意给反馈是吗？

【答】

就如果我觉得不好我就不说了。如果他有什么问题或者是他写得好我肯定会给他点赞，就给他一些夸赞，那有什么问题我也会直接问，如果他那个还比较活跃的话，经常还有人提问有回答的话。如果觉得不好一般就不说话，关掉就好了。

【问】

那什么因素促使你来做这个评价，做这个反馈呢，你的动力是什么？是希望它变得越来越好吗？

【答】

对，因为他本身就是开源的一个社区，每个人都是其中的一份子，那我也可以通过我的鼓励或者我的问题，让这个社区内容越来越丰富，而鼓励可以促进这个发布自己内容的人得到一种精神上激励，从而让他更加去拥抱这个开源社区，让这个社区越来越好，而我本身，我一直要用这个社区里面的东西，我也要为他做贡献，可能跟它是非常强的利益相关的，所以当然我也愿意为了这个社区的利益去夸他两句。

【问】

那我们今天的访谈就结束了。