**李东雷**

2021-10-2 10:0:17

【问】

我们开始吧，嗯，你先简单的说一下你的专业背景，还有你做过的开发的经历了，就是你做过多长时间的开发。

【答】

我因为我本科是学计算机的，所以我大概从大二开始就陆续接到一些开发的，就开始学习一些开发的项目，当时一开始学的话也主要是从Java开始做起，做一些，当时以课程项目为主，后来就从课程项目逐渐，就是我们几个，比如说几个同学组成一个小团队，会做一些小的一些开发，然后后来，到大四应该有两年多的时间。两年左右吧，然后考研那一段时间就停了一段时间，然后到研究生之后就跟导师做了一些，当然也是刚开始做，然后有一些校外的一些，有些机构可能需要一些这个计算机方面一些软件或者程序，我现在也在帮他们，可能会做一些系统方面的一些东西。

【问】

是指校外的就是公司实习的吗？

【答】

哦，不是，是有一些，比如说它有一些机构，比如说培训机构，它可能并不是，它不需要去找一个专业团队去帮它维护一个比如说那个官网，或者说它要开发一个这个招聘系统的时候，它可能就是没必要去找一个大团队，因为他们那个预算也没那么多，我们就会找一些小团队或者找一些专业的人员帮他们做一做，就这样子。

【问】

嗯，大概明白了，所以说你的开发的经历就应该还是有三年，就两三年这样子。那好的，了解了。然后我们这个研究的问题，其中有一个就是针对中国的程序员的，中国的开发者的，然后我们想了解一下开发者的特征，，简单的问一个问题吧，就是你觉得程序员这个群体和其它专业其它职业的人相比有哪些特征，有哪些不同的地方？

【答】

这个我感觉，大家都知道一点，就是年轻嘛，然后对于年轻人来说，他接受新的知识的能力比较强，然后他就需要不断去学习，这是我们区别其它行业的地方，计算机这个行业的技术迭代非常快。其它就跟计算机系统的有关，比如说他们写代码。就是我感觉中就中国这边写代码的话，他们就一般来说，可能在设计上花的时间比较少，他们在讨论论证环节，相对时间比较少的，相对于一些外企的啊，这会导致他们在软件实施的过程中，尤其后期的时候可能会维护，然后出现很多的困难。

【问】

嗯，您是说中国的程序员在，就是在真正的去实现一个功能之前是没有做太多讨论。

【答】

它可能就在调研方面可能做得并不是那么充分，像外企比如，就是说我听一些同学他们说，在公司内也是这样子说，就说他们做一个系统，比如说一个软件开发的时候，他们大部分的时间都是在团队做关于这个系统的论证，就是讨论上面，就说讨论一下这个系统是怎样，我们现在该怎样来实现这些东西，在写代码之前他们已经花大量时间在设计方面。中国程序员，大多数，我觉得有很多小团队来说，他看一个需求文档之后就直接去实现一些功能。他对代码方面的维护可能就考虑得不是那么周到。但这对后期比如说我们要重复一个系统，或者说，做一些拓展啊，可能会出现一些问题，然后遇到一些bug的时候，可能也会，这个东西不太好定位。

【问】

了解了，这个是你根据自己的经验来对比来得到的是吧？还有就是其它同学的经验？对。嗯，明白，那你觉得程序员这个群体，它从学习的习惯上来说有什么不同？

【答】

因为这个新的技术啊它出来之后它可能就没有，比如说我们像书本上的学习，它可能就是很多年前的知识，而且书本比较系统呈现知识，程序员就是，比如说一个公司，出来一个新的技术，比如说出现一个数据库技术，那可能有的公司就想要用上之后，因为技术刚出来，它没有那么多的教程，可能就更多依赖于这个开发者的文档，官方文档文档，他们在学习的时候，可能就像零碎的一个知识。就可能是用的时候，去查文档，像查字典一样。

【问】

就比较零碎，没有那种系统化，因为它这个更新迭代的速度，就是软件更新迭代的速度太快了是吧？

【答】

就是维护的也跟不上软件的进展，就会出现对不上的情况。

【问】

那这个是说文档本身，那程序员本身，就是从学习的角度来看，你倾向于是自学还是求助它人？

【答】

肯定是自学，占95%甚至更多的比重。因为通过文档来自学的话，它速度肯定会快一些。那求助他人的话，那肯定是自己解决不了的问题之后再去。但他们在运用新技术，比如说我们用一个新技术之前，我们可能会讨论一下。这项技术有哪些好处有哪些坏处啊？我怎样去快速掌握。

【问】

好的，那下面一个问题是，就是你可以大概回忆一下你最近学习了什么新技术新工具或者软件吗？那整个学习的过程是怎样的，就是我们希望你举一个比较具体的例子，就是说你从听说这个软件开始或者技术开始到开始用这个工具，学习这个工具，然后用它来做开发或者是一些功能的实现等等，就是整个过程是怎么样的。

【答】

我举个具体例子。那举一个Pytorch，因为我之前，我学这个技术的话也有，因为我之前做这个这方面，就是做那个深度学习的时候，用tensorflow比较多，后来发现tensorflow它从1升级到2之后，就是版本之间的差异比较大，然后它对，上一个版本就是说，这个2版本和1版本不兼容，然后就换到了Pytorch这个框架。我大概是从21年3月份开始去学习去使用，那我们学习的一部分就是上官网去查文档。它官网有一个60分钟的一个视频教学，我大概去浏览了一下这个视频，知道它这个Pytorch基本的架构啊，哪些模块我们要使用，然后我们就去查对应的官方文档。因为中国的开发者提供了一个文档，我们通过这个文档，我们主要看了前几章，第1个关于这个使用的一些细节方面知识，然后去查了一些中文社区上对于使用这个技术的一个总的一个概括，我们大概知道心里有数了，就是使用它的时候需要哪些步骤和注意哪些东西，然后在实现具体的功能的时候，比如说实现一个demo吧，实现一个demo，我们怎样去做数据加载，数据处理，然后怎样去构建一个模型，怎样去让它来训练，它每一个接口和API它都有很多参数，这都是我们通过查文档的方式来知道它里面的功能去实用，所以开发者文档对于新技术的学习是一个非常重要的一环。我们当时学习很顺利，主要是它这个文档感觉做得也，和版本的对应也非常好。嗯，那如果再举另外一个例子就是说，就是不好的例子就是，网上投诉还比较多，我们最开始开发这个微信小程序的时候，它就有大量的问题。因为微信小程序大概在18年左右吧，那会刚出来，那个文档也不是很全，但它提供了官方文档，通过下载开发者工具，然后去看文档一些demo的方式，我们去实现一个程序，然后发现那个官方文档上，就是同样的代码曝出大量异常，就是同样的代码复制过去就报错，对，报错，然后它那个文档维护的就非常的不好，然后网上吐槽也非常多，然后后来，后来我们就是通过上网查阅一些资料，那还有一些书籍，就是讲这块知识，书籍指的是那种教程之类的书籍，我们去查阅相关问题，然后网上也出现很多问题，相似的问题，我就是通过相似得方式来解决，但它那个文档最终，就是我们那个程序实现完之后，我们发现文档还是没有维护，也没有跟上。

【问】

你是说微信开发者文档？

【答】

微信小程序这个文档。微信平台上很多文档都是这样子的。

【问】

就相当于你整个学习过程还是主要求助于文档来学习，就直接使用官方文档来学习，对吗？

【答】

但它有一个前提，就是说你对这个技术有一个宏观的一个认识，已经有了一些基础知识，就说我要选代码，我要先做哪一块再做哪一块，就是流程是怎样的，你有这个流程之后，具体的流程细节之后，比如我们要做数据的加载，而我知道这一个大的快，那做数据加载，它可能要用很多的技术，比如说这个numpy，pandas，或者是pytorch它里面自带一些API。遇到这些细节的时候我们就查文档，遇到一些宏观的概念话，我可能更多的需要看文档前面一个概述部分，一个架构的总结也很重要，一方面就是别人实现过的一些经验，可能也比较重。它如果能提供一个完整的demo的话，是是非常的，这是非常好的。

【问】

那刚才你提到这个Pytorch，你看的是中文版是吧？对。这个中文版是它官方的中文版吗？

【答】

它官方就有中文和英文，可以切换。

【问】

然后还提到了这个社区，社区主要是指CSDN这种社区吗？

【答】

我查国外的社区比较多，因为像csdn，你刚才说是吧。它也有它的好处，就是对中国的开发者比较友好，但是有一个问题就是抄袭或者说这个转载就是特别多，然后你可能就往往可能找不到你想要的东西，所以说像国外的情况呢，比如说那个stackoverflow，它上面提供就非常的丰富，因为它那个受众可能也比较多，然后它那个上面各种问题的总结是非常丰富，你如果用英文去搜的话，在谷歌上去搜，就往往能搜到你想要的答案。因为这个社区这一块，像CSDN的话，他们做的我感觉更多的应该像一个总结，比如说，我学一个技术，或者过一段时间我再总结一下，然后我自己看的同时能给别人看，但它很多技术呢，它并不是那种提问方式。但stackoverflow它的这种提问方式，比如说我遇到什么问题，大家看看有没有方法帮我解决一下，然后后面可能会列出了很多方法，那它可能，它如果解决了，它有一个那个绿色的对号，那个CSDN或者像那个博客园，它更多像个人总结或个人做笔记的一种方式，就是做笔记然后顺便分享出来，也不管质量好不好，很多时候就是，你会发现一个链接链接到另一个文档，然后可能到最后发现都是同一个。

【问】

好明白了，我刚才提到了官方文档，提到了这种程序员的社区，然后还提到了教材教程是吧，那还有，就是你用到过其它的文档吗。就除了这些还用到，也不是说文档了，就是资料嘛，你平时还用到其它资料吗？

【答】

文档算一类是么？就所以的文档都算一类文档吗？

【问】

对，我们其实对这个开发者文档还没有一个明确的定义，然后现在的文献研究也还没有明确的分类，所以我们就想了解，开发人员他们平时在学习，在这个工作，在开发过程中有求助哪些知识，就求助哪些资料，这些我们都是需要了解的。

【答】

像有很多技术书籍，书籍它讲得比较系统，但它更多的是，它可能有宏观有微观，也有细节方面的一些阐述，但更多是想讲清楚一件事啊，或者说讲清楚一个技术，但你遇到，就实际上遇到问题的时候，我们在学习的时候可以这样做，比如学习新的技术，比如说可以参考这些书籍，但它书籍往往也比较厚，所以现在可能也不会去很想去看，然后在开发遇到问题的时候，主要就是比如说刚才讲Google搜索，然后像那个stackoverflow，发一些提问，再就是技术人员之间的一个讨论，也是一个重要途径，就是说可能有人之前遇到这事儿，省得你去网上在再去问了吧，直接给你解决，这也是非常重要的。然后就是，实际上文档里边它可能不一定说，只包含那种，就是说教你怎么使用它，我感觉正确的文档它那边也应该包涵一些，就是常用的问题的一些总结，Q&A，嗯，因为他们最后在发布的时候肯定会遇到很多反馈，他们对于反馈做一个总结，然后做一个比较好的回复，我感觉很重要，这种也包含在文档里面，其实也算。这种可能，相对说来比较少。就是说不是很多，更大篇幅可能在讲它的技术，这个怎么弄，这个怎么用？有哪些API的参数，这个参数是什么意思，对，就是这些。

【问】

就是说现在的官方文档里面它主要是这个具体的做法，就是说它大部分内容都是讲具体教你去怎么实现的，但是它没有预料到或者没有预想到你可能遇到的问题，然后它即使有Q&A，列举出的问题也不是很全面。那你觉得文档对开发工作来说重要吗？

【答】

毫无疑问是非常重要的一件事，就刚才说，就我们用一个技术的时候，不管它是新技术还是老技术，就我们跟查字典一样，我们实现一个功能。我们开发的人员啊，它可能并不需要去把所有的API都记住，这是没有意义的，我们遇到一个问题的时候，我们就查对应有没有解决的方案，或者说有没有对应的API，我们直接调用或者说去复用。它让我们遇到这种问题，就像查字典一样，我们有一个不会做，我们去查一下，查字典过程中或者查文档过程中，它要了解这个技术，它必须通过文档来了解，但它本身是不需要说啊，我遇到一个技术的话背下来，以后遇到之后就省得查文档，没有必要，就直接看文档就可以。

【问】

就相当于文档的起的作用，就是一个工具书，在遇到的时候查一下，然后这个文档它一是可以让你解决某个具体的问题对吧，然后另外一个就是让你了解这个技术。

【答】

而且我感觉开发人员更多的应该把时间和精力放在系统设计方面，要做有前瞻性，嗯，能够对未来说，扩展的时候啊，或者说做一些别的功能升级的时候，或者说做一些系统的迭代啊，做分布式调用的时候可能会，怎样去实现这些细节啊，或者说它在设计方面应该花更多的时间，而在实现方面，其实相对于设计来说是，相比较来说是少一些的。大型的系统或者比较特别好的系统，感觉应该都是这样子。

【问】

嗯，那您刚才说提到了就是设计比较重要，那开发文档对于设计本身有一定的意义吗？有作用吗？

【答】

我觉得设计方面的话，可能因为文档更多针对代码可能比较有效，但是设计方面的话，它可以有一些文档，比如说，在设计方面我们不同系统之间怎样来进行，不同技术怎样的组合或者说搭配，看这些宏观方面呢，可能对于一个，比如说它专门是做这个python的一个文档，它可能没必要去说，我想着以后怎样和Java进行相互调用，它就可以调用，通过这个资料，可以实现这个函数级别的调用，让它自己在做的时候可能完全是，可能想不到这些东西，它是想我怎样把我python的功能解释的更详细更通俗一点。我感觉设计方面更依赖于这个开发人员的经验和团队的协作这方面。

【问】

就是文档它更多的是聚焦在代码，具体的代码实现，所以对在开发过程中，在具体的实现上，文档可能起到更大的作用，嗯，明白。那你在开发的时候遇到问题是会怎么办呢？是会先看文档呢还是会去问别人？

【答】

我的习惯一般就是去谷歌先搜一下，看有没有对应的答案，然后如果说搜到了是吧，我就省得去文档页面查了，然后就可能直接解决掉，比如说我在第1页还没找到，我看这个，就是看文档上我对应的函数啊，或者各方面实现有没有问题，没问题的话，我可能在就是谷歌再去搜一搜，来换个关键字自己搜，然后比如说换个英文就搜这样子。

【问】

就是说主要还是通过搜索来解决。会有直接看文档的情况吗？

【答】

也有，比如说我之前用支付宝接口的时候，开发一个教育系统上面需要用支付宝的接口，它那个网上实现，因为我们开始做测试的时候，那个支付宝，那支付宝网上实现的文档就非常少，因为反正就是那么一篇，大家都在复制，那这时候就去搜一下那个阿里的官方文档，遇到问题，它那个文档我感觉很贴心的地方就是它直接给了一份实现好的代码，一个demo，让我们直接去看它怎样实现的，我们就可以做到，就是说我们改点代码就可以直接调用的情况。

【问】

所以觉得这个阿里的支付宝的文档比较好。嗯，那另外一个问题就是你阅读文档的时候，比如说你已经搜到了这篇文档，然后你点进去看，您是怎么快速的找到自己需要的信息？

【答】

如果是学习的话，我大概就是看它有几个章节。一般来说还有一些介绍啊，一些概念那种章节，然后我们再看这个章节讲的内容，总体宏观上也有了解，然后去细节方面啊，我们就学一些一些常用的功能。很多常用功能就是那么一些，当然一些生僻的比如说，就是我们平常不怎么用的，那会就用的时候再查一下就可以了，那遇到问题的话，一般来说，看一下目录上找对应的那个条目看一下，或者直接我们直接打开，因为一个官方文档，都是是有pdf的情况下，我们直接就搜就可以了。如果是在线的那种，那就只能一点点去，通过目录的方式，就找个对应的那个条目，然后去看一下它的那个阐述。可能对应到别的场景，然后再去看一下。

【问】

会通过关键词搜索吗？文档内部那种关键词的搜索，或者是文档它本身里面就有一个小的搜索引擎。

【答】

这种一般没怎么用，没怎么见过。

【问】

那你觉得，你有读过一些国外国内的文档是吧，那你觉得它们的质量怎么样？

【答】

坦白说，是国外公司做的更，总体来说质量更高一些，比如说我们用那个Java SpringBoot的那个文档，它官方的文档就是极其的详细，然后就内容非常丰富非常多，遇到各种问题，只要你从0到1把它念一遍，他都给你好了。国内文档，刚才说那个支付宝文档，我觉得还不错。其它很多，有些小公司，或者说，包括微信那个文档，它可能是做的不是那么好，或者说它本身代码都有一些问题，代码有时候就跑不起来，就出现这种问题。总体来说，因为这个国外的互联网发展也比我们要早一些，我们国内的感觉更多是可能还是处于一个野蛮生长的这样一个状态，高速增长可能就抢占市场，它在技术方面的重视其实没有那么多。但头部互联网的话可能还好一些。但国外的公司，很多公司它，大部分公司它的文档很全，然后那种小公司的文档，在小公司它就是一个小团队做出来的。它做一个小而美的一个系统，比如说我们用经常用一些ide，嗯，那个ide PyCharm，他们也是一个小团队做出来的，他们那个文档就写得非常全面。

【问】

就是说小公司做出来的文档也比较全面，能够考虑到开发者的这个需求。

【答】

主要是说我们国内像做基础软件给别人用的这种系统就比较少，嗯，我们国内可能能做数据库，比如说我们用mySQL对不对，那也是国外的一些，它一开始的时候被大公司收购，我们国内就没有这样的公司，比如说我做一个数据库给你用，也不是说没有啊就比较少。我了解一个PingCAP，它是做那个分布式的数据库。当然它的团队也不是很大，像这种公司的话它是有要求的，但是比较少，但是阿里现在也自研数据库的组件了，现在他们会有自研数据库，现在好像有一个时序数据库。反正是有的，但是相对国外来说没那么丰富，而且我们在使用的时候可能更多的用国外的，那个PingCAP那个，它可能就是对于比如我们做一个小系统，可能就是未必用到，它更多面向一些企业用户，可能比较大型的，像2B的。

【问】

那就是你刚才也提到了一点，像国外的这种文档它做得比较全，然后你除了觉得它做的内容比较全面，觉得还有做得比较好的，有哪些点。你觉得它这个文档比较好，除了内容好，还有其它的方面？

【答】

结构也是一方面吧。因为就是你写一个文档，就是结构加内容，然后结构的话，它有一个比如说一个社区就专门去讨论一些出现的问题，而且他们可能去直接去参与这样一个社区的一个活动。

【问】

是指在这个官方文档的站点里面就有一个社区吗？

【答】

也不是，他们当然有一个单独的网站，比如说一个社区，比如说Java都是有一些生态社区的，他们可能也可能也会参与到一些社区的一些问题当中，比如社区主要是讨论啊，或者说开发相关组件之类的，他们也会有相关的，那些国内这样做的，国内现在也越来越重视这个技术质量的，也做得越来越好，比如你说的那个平台。感觉他们，我没用过，因为我知道这个公司它做分布式那个系统方面它那个具体技术我也没有去使用过。但我感觉他们从网上评价看来，做得还好，还不错。

【问】

那就是除了内容、组织架构，你觉得像他们在做的比较好的，在交互设计上做得怎么样，因为文档它本身是一个文字量非常庞大的一个东西。比如说它可能会提供一些关键字搜索的功能，复制代码，甚至代码可以在那个demo里面直接交互实现。

【答】

反正有这样得一些，那个叫啥？开源文档，他们可以在那直接写代码，这种都是小的demo，它就像一个大的demo的话，它可能要跑一个系统也就跑起来，它可能你一个在线的可能就是满足不了这样的需求。

【问】

但是就是对作为程序员来讲，你觉得就是还是需要这样人性化的交互设计，对吧？

【答】

对，肯定是需要得，交互的话它我感觉比如说像UI它也属于属于一部分对吧，然后像那个搜索也属于交互得一部分，像UI那就是它设计的比较合理啊什么的。就有的文档做得烂的话就点进去之后啊，就真的就是在文档里面居然插着广告，做得就是非常的不尽人意。

【问】

能方便透露一下这个是哪个例子吗？

【答】

比如说我插一些，尤其是网上有一些，比如说，它应该不是官方的文档，有一些个人开发者发布的文档，像gitbook，像那个peros，有一个自己维护的，个人维护的pytorch，就是这种，很多这种文档是通过git book方式分享出来，它里面会插一些广告。

【问】

非官方的，偶尔会有广告，这个也是觉得你觉得交互方面不太好的，哦，那对语言你有要求吗？

【答】

程序员的话一般来说都是双语吧，虽然说它可能这个嗯，说和写方面可能不是很强，但是尤其像到了，或者它程序员接触专业词汇比较多的情况下，他一般来说是直接看懂这个英文的文档，因为英文文档，说到底它是一个技术文档，它里面的词汇，更多还是倾向于这个计算机方面的专业词汇，然后这个这方面的开发者，这词汇也是有那么一些，但他能看懂的，我感觉就我个人而言，我是大部分都能看懂。

【问】

就是你自己阅读起来没有太多障碍，那这个阅读速度？

【答】

嗯嗯，速度可能相当于中文，可能会慢一些。

【问】

那你有没有希望就是有那种中英文切换？

【答】

当然希望有这种，刚才我说的pytorch，一开始我就直接切换到中文就看中文，它如果中文和英文实现的，就是描述上是一致的话，并且做了一些加工的话，那看起来就会赏心悦目，而且也比较的，就是体验得好。它有的，直接拿谷歌翻译就比较敷衍人了。

【问】

就是你现在觉得做这个双语的比较好的一些例子，除了pytorch。

【答】

Tensorflow也算一个，像我之前说那个springboot当然只有英文的，应该只有英文，然后其它的，但语言类的话，那就直接选也没什么啊，像Java的话，很多都是英文的，没有中文的。因为外国公司它做中文的话还是比较少，他只有在国内特别流行的情况下它才会，或者它为了某一时期去推广这个技术，它不只是中文和英文，会做很多种语言。

【问】

那你能回想起一些比较差的文档的例子吗？除了微信还有其它的吗？

【答】

它差是因为本身它写的代码，本身可能就在我这边就跑不过，它本身就有点问题，然后当然不是我一个人说可能环境配错了，是网上这方面的声音很大的，很多。

【问】

就是除了这个内容本身，它的架构组织架构方面。

【答】

我感觉它那个界面啊，或者说说那个排版啊，或者它那个，因为它那个文档可能分章节。那个导航栏它做的还是可以的，就是那个内容方面有些问题，而且它因为它一开始出来的时候，它可能后面有些迭代，它那个技术已经迭代好了，但那个文档没有跟上去，就是版本不对应。

【问】

国外的有这种情况吗？

【答】

国外可能也会有了，它可能也不是说就是做的那么好，那我们国内其实现在也在跟进，现在做得越好，当然也不是说国外的就一定好，肯定有一些问题，但我遇到的，就是也不是特别多。

【问】

就是你遇到的国外的那种产品的官方文档都还是比较好的。

【答】

对，因为就像我用技术的话可能就更多的去，就为了这个，嗯应该说这个系统能够稳定运行，我们可能会做更多的，就说比较流行的或者这个用户量比较大的，因为这样的话它那个产品就是有一定的保证，它这方面就做的会完善一些，它可能小的一些，嗯，之前是那个fast test吧，反正就是有些技术，它那个文档也是有一些问题，可能也不是那么十全十美的，然后在国内也有很多好的。而且国内很多小团队它可能也会实现的，像做开源的话，它开源项目比如说它技术做得非常杰出，它的结构做的非常好，像阿里那个fastjson它也是有些文档，也做的挺好的。我前段时间帮一个外面的公司，帮一些机构做了一个OA系统的时候，用的是其实是我们国内开源的这些，它那个OA系统有一个文档也非常好，就说它怎样把这个系统跑起来，是实现哪些步骤就按照哪些组件，它都有专门的那个阐述，然后后面说你做一些定制化配置的时候，要怎样做，它都有详尽的，就描述得非常详细，然后后期你还添加一些自己的组件，都是可以的，那它还有些Q&A。

【问】

你认为Q&A在这个官方文档里面其实是非常重要的。

【答】

对，也非常重要，你遇到问题的时候，就即使你没遇到问题，你可以看一看，看一看后期可以避免，尤其写一些代码的时候，你看一下，可能就可以避免这个问题，就在用之前就给看一下。

【问】

那除了这个就是，接下来是想问一下你写过技术文档吗？做了这么多的开发，有自己写过文档吗？

【答】

我写过那种就是它也不算技术文档，比如说系统交出去的时候，更多是像用户手册，怎样去使用。因为我们这个团队也没有做过什么，就说，嗯，能够直接给开发者使用的一些组件，所以说这方面可能比较少，但我感觉那个像文档吧可能也不一定说，只包含我们指定的文档，可能看一些开源代码的时候，那个注释可能也算一种文档吧。

【问】

对，你提到的非常重要的一点，就是注释也算一种文档。

【答】

而且像JAVA很多那个文档，它可能就从注释里面提取出来的，它能够直接组合。那如果做开源项目的话，或者说把这个代码贡献出来的时候，注释我感觉是非常重要的。

【问】

注释就是在代码里面直接写的那个对吗？

【答】

对，当然是在代码里写，它在那个就是我们写一个函数的方法的时候，我们的方法前面，要去阐述这个方法有没有用，它要传输哪些东西，会返回什么类型？然后我们对我们定义的一些变量或常量，你要做一个详尽的阐述。这个不不写注释的代码是很恶心，那你就一点点就一点点去看去想，你很长时间去想这到底是想干嘛，这就很繁琐的过程。注释写得好的话，对自己对别人都有一种就是说非常省事，一个高效的方式。

【问】

除了刚才提到的这个是Java是吧，提到了Java，还有哪些就是它提供的demo里面注释写得比较好的。哦，你说的是开源的是吗？

【答】

开源项目可能得依赖开发者嘛，而且这个背景最大的开发者，他可能宁愿多写几行代码啊，他也不想去写那个注释，写注释是一个很烦躁的过程。而且就是，我感觉就是有很多的开发者了，尤其是说，因为大公司它可能一套规范流程，所以你必须要你写个代码，可能必须要去写注释，这都是有这个硬性规定，他们可能就会去写，像一些小团队啊，或者说，有些大学生在做一些项目的时候，它就不想写注释了，因为它那个整个系统做下来可能就没多少注释，那它如果想把这个开源拿给别人用的话，就我拿过来用吧，我要去跑一遍，然后去看它那个代码到底什么意思，所以是一个很烦的过程。然后我最近也在看阿里开源的一个组件，那个rocket and call, 消息队列组件，它的代码，那个代码里面注释也写的也是非常详细，而且它现在做的非常好，因为它主要是想你，它做一个开源组件，可能这个用户量非常大，很多公司在用。而且开源有用户用了之后就会提升它这个公司在技术方面的影响力的，他们也是提供影响力，想推广，这个技术文档，它有那些技术文档，他的文档有中文的英文的，然后文档对他这个整体的这个系统的架构和细节方面也有很多阐述，而且在代码里面，那个代码，注释是非常，就是我感觉是非常完善的。但他的注释也都是英文的。

【问】

就是说如果这种官方的产品，官方的文档，它做开源的话，它应该是有一个比较规范的规定，然后写出来的注释也比较规范，但是像一些个人那种开源，就比较小规模的开源的的组织的话，可能写出来的注释就是比较参差不齐的。

【答】

对，这对开发人员说写注释真的是非常让人不喜欢的一件事儿。

【问】

那最后一个问题就是想，因为我们这个项目是跟企业合作，是想帮助他们评估他们自己写的这个文档的质量，开发者文档的质量，然后帮助他们提升质量的，然后我就想问一下程序员，你觉得什么样的文档才是好的技术文档，就除了刚才你提到的几个比较优秀的例子，他们的优点以外，你还能想到其它什么吗？大概总结一下，总结几个点。

【答】

一方面的话嗯，做一个技术文档，它要把这个技术就是拿给别人用，用英文说就是get started，就是开始第1步，总结一个，就你这个系统最好有一个demo，就是你能给开发者提供一个能跑起来的一个案例，然后它后面，如果能直接跑起来的话，它后面就是做了很多的比如说中文和英文的阐述，它都能和你代码去对应上，读起来就轻松一些，如果说直接没有这个demo或者没有这些案例的话，那么它，应为看文档是件很枯燥的事儿，然后它也就没法去验证你这个到底对不对，它在它自己系统去验证的话，可能它专门去开一个新的一个demo一个工程。

【问】

就说第1点是要有具体的例子，能跑起来的例子。

【答】

第2点的话就是，就是我们说那个导航栏，它那个文档还有一个科学规范的一个组织，对它要科学规范组织你这个信息，就是信息全放到一起的话可能就找起来就非常的麻烦。然后第3点的话我感觉对你这个系统不同组件，因为你一个文档拿给开发者用的话，如果拿给用户用的话，那就是算普通的说明书，你会怎样去使用它，用户就拿它那个系统，它就黑盒了，他不需要知道你这个有什么用，我只要点这个按钮，能出现什么东西，知道这些就够了。但对于开发者来说，它肯定就是去看里面的细节，技术实现细节的话，你这一个函数或者一个类有什么具体的作用，然后这个函数它返回的类型，返回的参数它有什么作用，或者它里面发生了一些什么，就是怎样去实现。

【问】

就是说你不光要教我怎么怎么用，还要教我它为什么要这么用，就讲一下原理。

【答】

对，讲一下里边的原理。但这是针对啊，比如说我们提出了一个函数。还有一个就是你这些代码组织，你怎样组织代码的，比如说你，因为程序员有一些UML那个工具给你画一些，通过画图的方式，你讲一下你这个不同组件或者不同类之间是想怎样组织起来的，然后对程序员对开发者，他来定位或者来学习的时候都是一些帮助。

【问】

就是说除了文字也需要有一些图片说明。

【答】

对，图片说明，图片来辅助对这个系统的讲解、理解。然后，当然这些都是宏观方面，最重要的是你这个代码的阐述方面一定要详尽，一定要得到，然后就是不能太少，太少肯定是阐述不清，你也不能有原则的错误，有些它可能就是实际上有错误，而且你提供出来的代码，刚才说的那个案例对吧，它是一个完整的系统，它可能会后面也会有一些小的一些代码，行数不多的一些小代码，这代码你一定要确定，你这个代码在系统，就说我去拿来用的时候，它没有什么原则性的错误，嗯，这是正确性的，正确性方面，然后还有就是UI，UI交互体验。就你一个企业，一个团队做一个文档的时候，你用不用心，可能就在你这个文字这个阐述方面，当然大家都重视这个方面，UI方面也是一个非常重要的一个点，就像刚才说的那些搜索，它能提供一些关键词搜索，这个我感觉在大部分的那个，中国其实是有的，有些会提供一个小的搜索框，很多时候它有的也没有，只能通过导航栏去找，有的他直接给你一个pdf文件，你就一点点去看，就是交互上来说没有这种网站好。然后的话，我感觉它那个文档，嗯，其它的那几个刚刚想了一下，想不起来了。

【问】

还提到了一个这个语言的问题，双语。

【答】

对对嗯，中英文这个切换对吧，那在计算机语言上要，就是主流的语言，应该支持，C++, Python, Go, 就是如果你做一个开源组件来给别人用，不同语言，因为计算机语言有很多种，可能有上百种语言，你不需要去实现所有，但主流的语言就是大家都在用的语言都要去支持。

【问】

就是说这个软件它本身要支持不同的语言，然后它这个文档也应该有相应的，就是针对不同语言的这样的一个说明。

【答】

嗯，就在实现方面，因为文档它要，我们程序员看文档，它最终要落到实现中去。他实现过程中有些问题的话，他应该提供一个，比如说一个官方技术论坛，通过论坛的方式去提问，然后有不管是别的一些技术人员在上面给你讲解也好，还是说你这个团队的支持也好，就说你能解决这个问题。一个技术论坛，就像很多的社区，对吧，它很多技术要做社区的，因为社区做的好，对你这个开发者也好对你这个技术的推广和产品的迭代更新也是非常有好处的，因为通过不同用户来用，他就直接发现你存在的问题，帮你迭代对吧。

【问】

这其实也是另一种类型的Q&A，只不过是不是那种预想的Q&A，就是大家真的是在实际开发过程中遇到的Q&A。

【答】

对，然后就刚才说了一个点就是维护一定要跟上，不能说你版本升到了5，这个文档还停留在2，因为它系统更新的时候它可能有一些接口或者有一些代码会改变，或者它的参数啊可能会改变，或者它的名字就变了，因为代码如果跟不上的话就直接就对不上了。就网上怎么说，就说你是一个本来好的就是用了之后变错了。其它方面的话，像做的好的文档的话，我感觉它如果能提供一个视频的教程也是非常重要，比如说像那个MongoDB那个数据库，它就会定期的开设一些课程，就是专门讲解它这个数据库怎么使用，和Python系统和Java系统和c++系统对接去使用。然后它可能一方面去讲他们这个产品怎么样，然后通过别的一些产品组件的方式搭建出来一个完整的一个小型系统。

【问】

这个视频教程是放在它的官方的那个文档中心里，还是通过其它的视频网站链接？

【答】

一般都是它有它自己的官网，放在它自己的网站上，然后它那个文档的话，就说它文档第1个肯定是首页，或者介绍，一个introduction，那方面就会有一个链接说啊，我们这有一个视频的教程，然后你点我就可以去看一下。

【问】

挺好的这个点，你还有其它需要补充的吗？

【答】

嗯，大概就这些了，我就感觉一个文档它主要的架构方面可能就这些。就是一个团队啊，嗯，刚才我就说，它写注释就说，程序员可能就不喜欢干这事儿，它写文档也是一件这样的事儿。所以说作为一个团队能够通过一些制度措施和一些规范去保证这个文档质量也非常重要，因为它不想写的话，它就算写出来可能就是去简单去描述这个东西啊，为啥，这个怎么用，也就是这些东西。

【问】

就是现在的互联网公司，他们的文档团队都是叫一个混合的团队，就是既有这个技术人员，也有专门写文档的，就是一些文科背景出身的人。

【答】

好，那就非常好，像头部一些大公司的话，那他们要做一个技术文档，就尤其阿里的技术文档在这个国内是非常大的，就像Java一样，他们的文档可能做的肯定是会要好一些。

【问】

还有其他需要补充的吗？

【答】

大概就是这些了。