16 | 怎样才能写好项目文档?

2019-04-02 宝玉 来自北京

《软件工程之美》

你好，我是宝玉，我今天分享的主题是：为什么你不爱写项目文档？以及怎样才能写好项目文档?

我以前看过一个投票，盘点程序员不喜欢的事，有两条和文档相关：

> 不喜欢写文档;
> 不喜欢项目文档太少。

看起来很矛盾，却很现实。基本上大家都认同：“项目文档很重要”，然而我们在项目中总是**短期高估文档的重要性，而长期低估文档的重要性。**

结果就是口号喊的很响：要重视文档、要写好文档、要多写文档，然而随着项目的推进，总有比文档优先级更重要的任务，文档的优先级总是被有意无意推迟，导致项目的文档缺失、老旧、无人维护。

那么为什么程序员都不爱写文档呢？我总结了一下大致有下面这些原因。

不知道怎么写

不知道怎么写文档的应该占很大一部分比例。

太忙没时间写或者懒得写

程序员确实很忙，但总有不那么忙的时候，却也很少见有人利用这时间去写文档。包括我自己也这样，有时候没那么忙的时候，宁可去想想怎么重构下代码，却很少会愿意去写文档，主要还是太懒。

因为是敏捷开发，所以不用写文档?

对于这个问题，我其实反驳过多次，敏捷宣言最后一句话明确指出：“尽管右项有其价值，我们更重视左项的价值。”也就是说敏捷从来没有否认文档的价值，只是更重视“工作的软件”罢了。

为什么要写文档?

写文档，其实对个人、对项目、对团队，都是非常重要的事情。

帮助写文档的人理清楚思路

我想你应该有这样的感受：写作的过程，就是一个思考的过程。

写文档，可以让你在写代码之前，梳理清楚思路，想清楚整体结构，比如说有哪些工作是重点难点；哪些要依赖其他人，需要及早协商的；哪些是要考虑安全性的。

如果上手就写代码，就很容易陷入到某个技术细节中，而忽略了整体结构。写的时候才发现一个技术难点无法解决，或者已经在某个不重要的细节上浪费了很多时间；或是发现有些依赖其他人提供的服务还没准备好；又或者是上线后才发现有安全漏洞。

先写文档，就会抛开代码细节，去站在全局思考。写的时候，各个模块之间的依赖关系、各种可能的安全隐患、各种可能需要其他人配合的地方，就都冒出来了，必须要去查资料，去找人讨论，反复缜密的思考后最终写出来。

有人觉得自己写作不行，所以不会写文档。写作不行，只是让你在用词遣句上会有所欠缺，而这不是写文档的真正障碍。

真正的障碍是没想清楚，在心中只有一些未成型的混乱的想法和概念，必须要努力把这些模糊的想法确定化和具体化，才能写出来。

换个角度来说，如果你连文档都写不出来，那又怎么能指望代码写得好呢？

便于未来的维护和交接

“好记性不如烂笔头”，存在脑子里的内容是不可靠的，一个正常的项目组，如果需要长期维护，就需要一定的文档，把设计、操作流程、环境配置等内容记录下来，而不仅仅依赖于口口相传。

我有一个习惯，每到一个新项目组，就会把日常工作中遇到的问题、各种环境配置、一些操作的步骤等，所有以后可能还会用上的都记录下来，其中一些还会整理到团队的 WIKI 上。

一段时间后，这些随手记下来内容都会发挥大作用，对于我来说，很多问题就不需要问第二遍了。对于团队来说，随着人员更替，我记录的这些内容都是最好的一手资料，有新人过来，按照我当初记录的内容，就可以快速上手。

便于团队更好的协作沟通

在一个项目组中，大家都有不同的分工，有人负责产品设计，有人负责架构设计，有人负责测试。而文档，就成为了团队成员很好的沟通工具。

比如说产品设计有雏型的时候，会有一个产品设计的评审会议，基于文档，项目成员可以一起参与其中，提出自己的意见和看法。这样就不至于等到产品设计出来之后，大家才对于设计有

改进想法或意见，造成无法更改的结果。

当然，写文档还有很多好处，在这里我就不一一列举了。

如何写好文档?

其实文档的重要性真不用多说，很多人也不是不爱写项目文档，而是不知道该如何写好文档。所以在这里我来介绍一下该如何写软件项目文档。

很多人对于写文档是有心理压力，觉得自己写作水平不高，不知道该如何下手。首先你要对文档有一个正确的认识：文档写作，关键是通过文档把你的想法表达出来，至于用词、格式相对都是其次的。

打个比方，我们如果是大厨给餐馆做菜，得追求个宽油大火、色香味俱全，自己在家做饭，就没那么多讲究了，填饱肚子是第一要素，在这个基础上味道好一点就很好了。

我们写文档就像是在家做饭，是不需要追求太多华丽的词藻，也不需要追求字数，只要用简单的文字、图表把想法表达出来，最终在讲解的时候，配合一些口头说明就可以啦，其实比我们上学时写作文容易多了。

下面给你介绍一些具体可行的文档写作方式。

1. 从模仿开始

前面有提到，我其实一开始是不知道如何写文档的，直到毕业两年后，我在飞信项目组，领导让我写一个新项目的技术方案文档，我两眼一抹黑说不会写呀，然后领导给了我另一个项目的技术方案文档，说你就“照葫芦画瓢”照着写吧！

“依葫芦画瓢”就简单多了，同时又学习了一下如何画线框图、时序图等图形，很快就完成了一份技术方案文档，再反复修改几次，质量就还可以了。

后来我带团队时，让团队成员写文档，就把当时我写的文档给他们参考，很快他们也能写了。包括后来我写开源项目（angular-ui-tree, react-video），要写英文文档，也是去找了

几个同类的开源项目的文档，参照他们的内容和格式，就把文档拼出来了。

模仿就是最好的写文档方式，尤其是现在网上可以参考的例子也很多，当你写文档不知道该如何下手的时候，不妨去找一个类似的文档，模仿着写试试。

2. 从小文档开始

一开始写文档，内容不需要很多，可以从小的文档开始。就像前面我提到的，记一些笔记，不要在意格式，一两句话，一些截图，就是不错的笔记。

有一次和同事一起去开会，会上他给另一个组的人介绍了如何调用一个服务，介绍的很详细。我就建议他把刚才介绍的内容写成个小文档，这样下次再有类似会议就可以基于文档来说。

于是他就整理了一个简单的文档，再为别人讲解的时候就基于文档介绍，容易很多。同时，他每次还会再完善一点内容进去。之后再有同类问题时，他直接把文档发给人家就好了，都不需要再专门开会。

项目中很多文档都可以从这样小的内容开始：别人给你讲一个问题的时候记录下来；你给别人讲一个问题的时候记录下来；解决一个技术难题时记录下来方案……

这些记录下来的笔记，稍加整理，就可以是很不错的项目文档。

3. 从粗到细，迭代更新

小时候写作文，老师给的最多的建议就是要列提纲，这个建议我小时候当耳边风没怎么听，后来要写项目文档的时候用起来反倒觉得非常实用。

我写一个大一点的文档，都是从脑图开始的，先基于脑图，把基本结构梳理清楚。然后第二步就是写 PPT，PPT 有个好处就是不用太多文字，列个一二三，画几张图，就是个简单的文档，PPT 还有个好处就是可以用来给别人讲解，收集反馈。

写完 PPT，也收集好了反馈，再写正式的文档。先按照脑图列的提纲把主要章节放上去，然后把 PPT 上的内容和画的图放到文档中，一篇文档的骨架就搭好了，剩下的就是对细节的补充了。

为什么我不一开始就写很细的文档呢？

一个原因是太难写，要花很多时间精力，甚至可能写不下去；另一个原因就是在收集反馈的过程中，会有很多修改。**写得越细则无用功越多，最后，你甚至会因为不想改文档而抵触不同的意见。**

而从粗到细逐步迭代的方式就好多了，一开始的目的是为了梳理清楚思路，只要脑图这种级别的内容就好了，然后进行调整。因为文档很粗，调整也方便，等到基本确定后再写细节，就不会有大的反复。

4. 一些基本的画图的技巧

有人说：“字不如表，表不如图，一图胜千言”。这个观点我非常认同，好的图能帮助你简单而直观地把问题说明清楚。

画图其实不复杂，不需要多专业的绘画技巧，也有很多工具软件可以帮助我们简化操作，像 Visio、PowerPoint、Keynote、OmniGraffle 等都是很好的画图软件。平时看到好的图也要注意收集整理，以后自己写的时候，也可以直接参考，可以帮你少走弯路。

写文档的时候，主要有几种图比较常用：线框图、流程图、时序图、各种格式的截图。

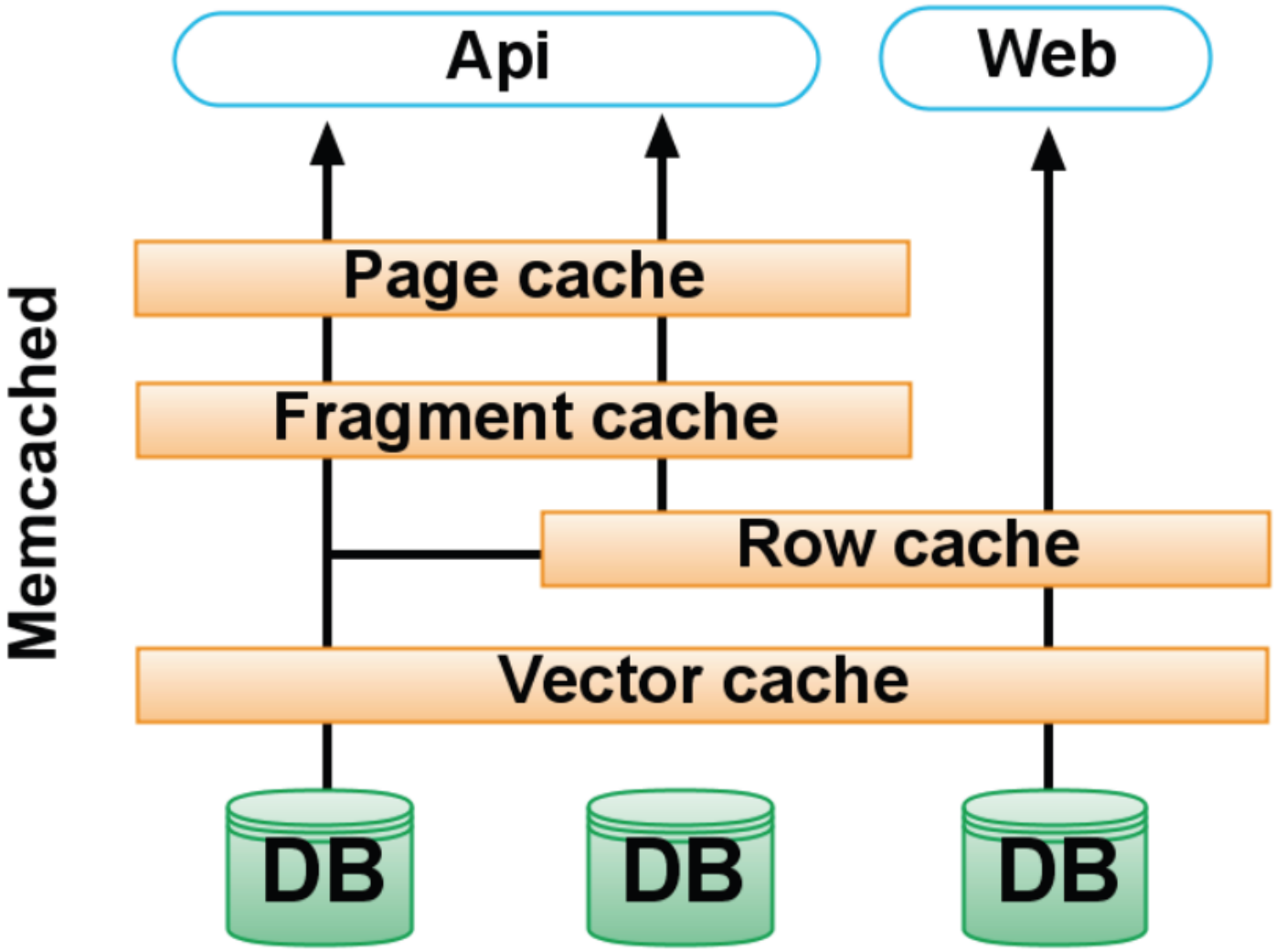
线框图

线框图是最常用也最实用的一种图形，用简单的方框代替功能、模块、服务等，再用箭头表示关系或者数据流向，非常简单直接。

要画好线框图并不难，主要是要理清楚有哪些模块，以及模块之间的关系是什么。用方框配上文字表示模块，方框之间的连线和箭头表示关系。

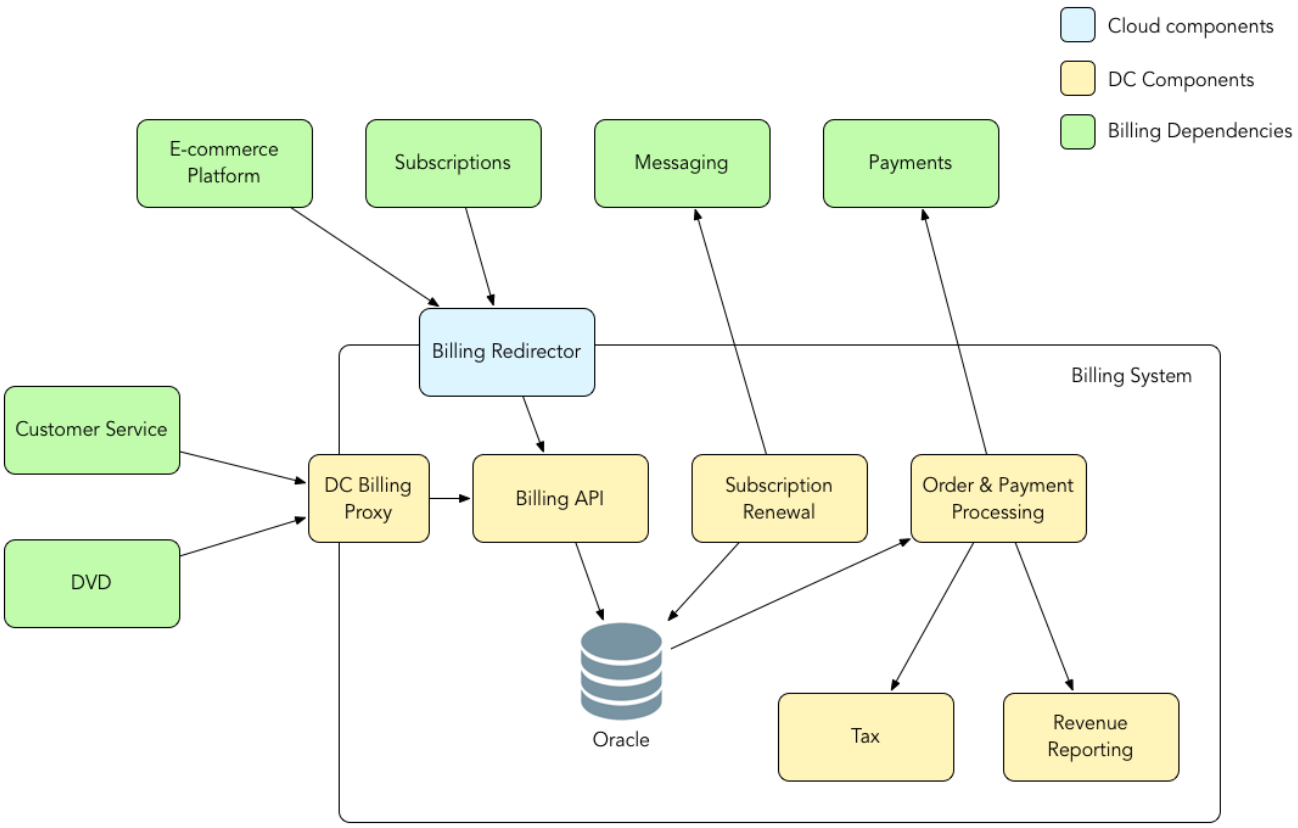
看几个例子：

例：Twitter 当年的缓存方案。



图片来源：InfoQ

例：Netflix 的账单系统架构图。

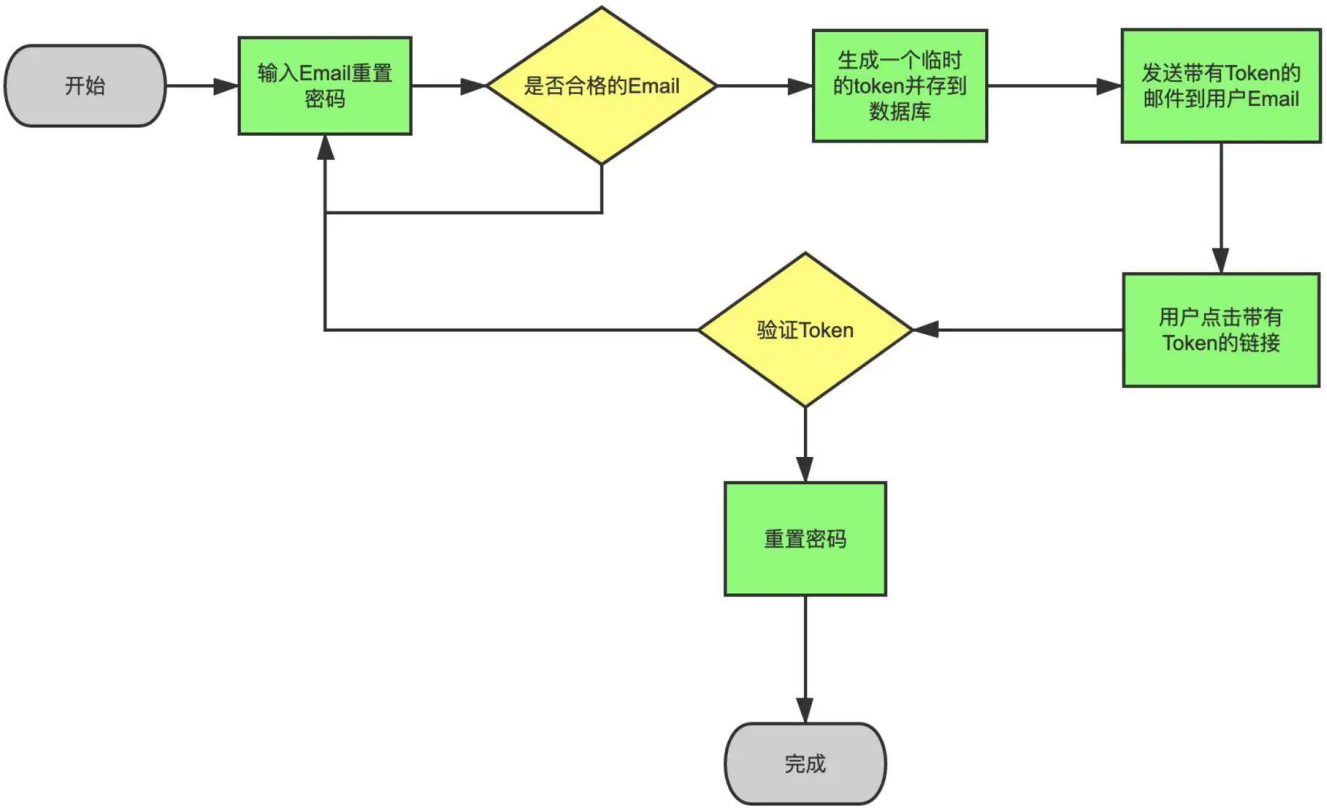


图片来源：Netflix技术博客

流程图

流程图是软件项目文档中一种常用图形，可以方便的表示各种不同条件下的逻辑路径。**要画好流程图不难，重点是要理清楚逻辑关系，各个关键节点在不同条件下的走向。**

例：重置密码流程图。



时序图

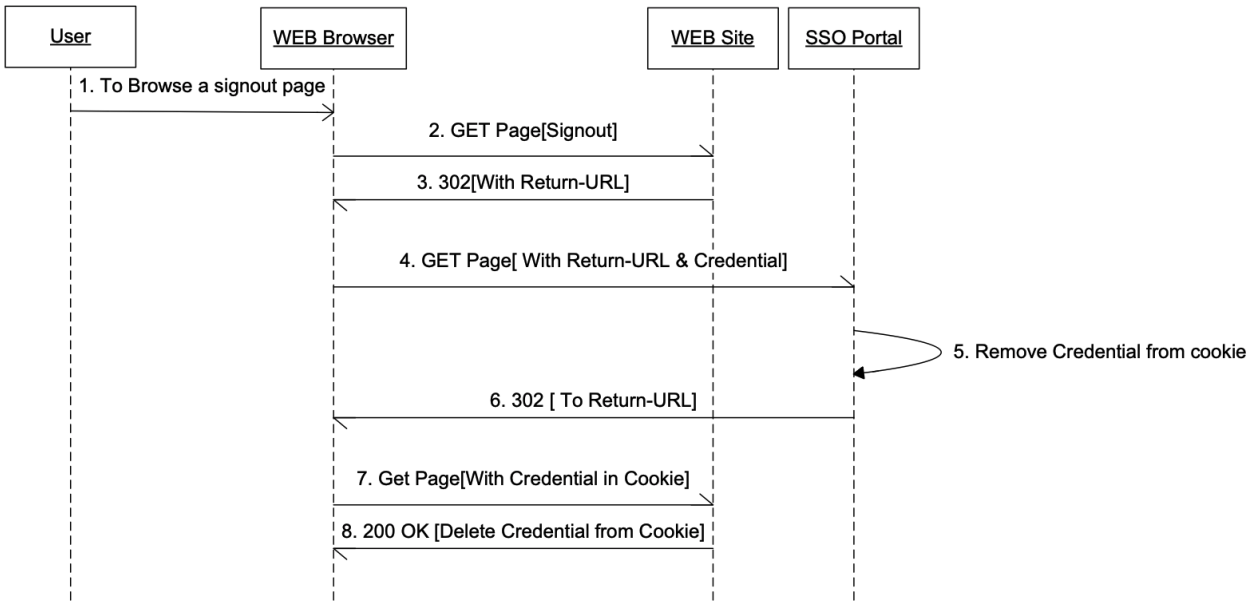
时序图也是软件项目所特有的一种图形，可以表示不同对象之间发送消息的时间顺序，尤其在涉及网络通信的文档中特别常用。

画好时序图，关键是要列清楚所有涉及的对象或者服务，以及消息发送的先后顺序。

例：注销登录过程的时序图。

3.3. 注销接口

3.3.1.序列图



1. 用户要访问一个 WEB Site 的注销页面。
2. WEB 浏览器向 WEB 应用发送一个 HTTP 的 GET 请求。
3. WEB 应用对 HTTP 请求进行检查，发现用户要进行注销操作，返回 302 重定向的应答码，并返回重定向到 SSO Portal 的 URL，应答中包含指向 WEB 应用注销页面的 URL。
4. WEB 浏览器向 SSO Portal 发送 HTTP 的 GET 请求 ,请求中包含授权凭证和指向 WEB 应用注销页面的 URL。
5. SSO Portal 将授权凭证从 Cookie 中删除。SSO Portal 返回应答码为 302 的 HTTP 应答，应答中带有指向应用注销页面的 URL。
7. WEB 浏览器向 WEB 应用发送一个 Cookie 中带有授权凭证的 HTTP 的 GET 请求。
8. WEB 应用将授权凭证从 Cookie 中删除，并向 WEB 浏览器返回应答码为 200 的 HTTP 应答。

各种格式截图

截图也是个非常简单直接的方式，把软件的 UI、交互设计的效果、数据趋势图、数据统计图等直接截图，必要的话配上一些箭头、文字，也可以很好的说明清楚问题。尤其是产品设计文档，经常用到。

上面就是如何写文档的一些具体建议，按照上面说的方法做，写好项目文档不会是多难的事情，你还可以在日后的工作中，不断学习不断改进。

一些关于文档的其他建议

有时候我也看到一些比较极端的情况，就是过于追求文档，项目中要花大量的时间写文档，而很多文档是形式化的，并没有太大意义，可能写完了不会用来讨论，也不会有人看。

所以我是比较认同敏捷宣言观点的：文档很重要，但是工作的软件高于详尽的文档。这里面的平衡很重要。

不需要为代码写很多文档，好的代码格式，良好的注释、完善的单元测试可以很大程度上代替针对代码而写的文档。

Markdown 是一种非常好的文档格式，可以让你更专注于内容上，而不是文档格式上面。

在线文档工具优于离线文档工具，在线文档有很好的版本管理，也更方便多人协作。像 GitHub WIKI、石墨文档、Google Docs、Evernote 等都是非常好的在线文档工具。

对于文档的撰写，要作为一个正规的项目任务进行，安排人、安排时间，放到项目计划中去。就像前面说的“懒得写”文档的情况，一旦把文档当成一个与开发同等重要的任务去执行，就没有借口去犯懒了。

重要的是，文档的写作一样需要多练习，写的越多，就越熟练。

总结

今天，带你一起分析了为什么不爱写项目文档的原因，也解释了为什么写文档很重要。

没时间写或者懒，不能成为不写文档的理由。对于重要的项目文档，就应该加入到日常的开发任务中，把写文档，摆在和设计、开发同等重要的位置。从某种角度来说，写不好文档，代码也很难写好。

针对程序员不爱写项目文档的情况，我也提出了切实可行的写文档的方法。比如说不会写，就可以从模仿别人写的文档开始，然后从粗到细，不断迭代，配合一些图表，就可以写出不错的项目文档。

课后思考

你所在的项目组，项目文档情况如何？你写文档吗？有没有什么写文档的经验和大家一起分享的？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。

精选留言 (29)

青石
2019-04-02

领导常说"大脑是用来计算的，并不是用来记忆的。"

工作年头越多，越习惯将平时操作的过程整理成文档，分享给内部成员。

学的东西越多，记住的内容往往越少，将重复或可整理的内容写成文字，保存起来，使用的时候知道去哪里找就好。这也正是索引/缓存的妙处，利用大脑有限的Cache资源缓存常用的内容索引，将不常用的内容存盘，需要的时候再次加载。

作者回复: 👍严重认同!

大脑是用来计算的，并不是用来记忆的，记忆一个索引就好了。

共 2 条评论 >　　53

易林林
2019-04-02

项目文档是整个项目的骨架，皮肉器官得自己一点一滴的增加，这样可以做到有迹可循，有法可依。开发人员写文档和写单元测试的心理是相似的，自己不愿写，但抱怨别人写得少。直到离开公司的那一天，总结起来就是业务代码写了一箩筐，还不能清晰的理清整个项目的业务结构，CURD的操作都赶上一万次理论了，但CURD怎么配合使用最高效、底层原理是什么，一概不知。

宝玉老师讲到不会写的问题，这确实是大多数底层开发人员的通病。有的时候很可气，比如邮

件格式错误、文档有部分错别字、逻辑表达不清等等，都会让看的人很抓狂。在我看来，只要是合格的开发人员，都应该通过刻意练习来提高自己的软实力（英语水平、写作能力、语言表达能力等），不能只在乎自己硬实力多么多么精湛，让周围的人感受不到可以学习的空间。

部分开发人员到一定阶段会有很多的困惑，典型的就是努力的学习技术，却好像离技术越来越远，有一种恐惧和迷茫。其实，这是需要去克服的瓶颈期，输入（学习）的东西很多，输出（表达）的东西很少，学十只能知其一，学一百知十...，就会发现会的越来越少。相对于只进不出的一潭死水总是没有涓涓细流走得长远。

开发人员个人分享、写文档、写技术博客等等，这些可以伴随你走得更远的软实力都是必不可少的一门学问，甚至超过了技术本身，当然，这只是个人感想，也还处于摸索中。

> 作者回复: 说的太对了，还有单元测试也是一样的，自己不愿意写还抱怨别人不写😁
>
> 后面几段总结的特别特别好👍
>
> 写文档比写作容易，都不需要刻意练习，稍微用点心就可以不错了。
>
> 输出就是要靠写，靠教，靠悟。就像学习攻略里面讲的：用器、学术、悟道、传道。

  20

Bo
2019-04-03

已经写好项目文档，但想更另一步优化文档，老师可以分享一下项目中需求规格说明书，概要设计，详细设计，代码规范文档，测试文档，部署文档等的优秀具体案例吗？特别想看看老师的，模仿学习？

> 作者回复: 有些内部文档不方便分享。
>
> 我在文中附了一个开源项目的链接：https://video-react.js.org/
> 这个是一个组件使用文档
>
> 其实类似的有很多开源项目的文档都写得很好。比如：
> Vue: https://cn.vuejs.org/v2/guide/
> Redux: https://redux.js.org/
>
> 还有可以网上搜索一些，例如：

产品需求文档模板：https://www.jianshu.com/p/e89e97858be1
微服务：从设计到部署：
https://docshome.gitbooks.io/microservices/content/2-using-an-api-gateway.html

16

hua168
2019-04-02

老师，你能简单说一下项目前-->项目中-->项目完成，一般都需要哪些文档呀，有没有示例或链接或搜索关键词？好让我们没有项目经验的，能见识一下

作者回复: 我大致列一下，可能有遗漏的：
项目立项：

原始需求文档
可行性分析报告
立项说明书

需求相关的：
原型设计文档
产品设计文档

系统设计相关的：
技术方案文档
详细设计文档

开发相关的：
代码规范文档

测试相关的：
测试用例
测试验收报告

运维相关的：
部署文档
故障报告

其实没有特别的标准的，还是根据各个阶段需要。原则上就是：

1. 这件事需要讨论需要评审，要有文档作为讨论的依据，以及记录讨论的结果。比如各种设计文档
2. 这件事要有规范，要有文档保证规范统一。比如各种规范文档
3. 这件事要记录下来，作为以后的一个参考。比如各种报告、环境配置、操作手册、API文档等

14

hua168
2019-04-03

谢谢😄……那不是在写文档上花费时间多，他们不一定会看，主要是为了以后方便维护和修改吗？

作者回复: 磨刀不误砍柴工！
写有价值的文档、写单元测试，看起来要"浪费"一些时间，但是长远看，可以节约大量的时间。

5

dancer
2019-04-02

文档的好坏也能看出一个程序员的水平～另外把文档也算做任务中一部分这个主意很赞，是不是也可以当作一个ticket？

作者回复: 🤝英雄所见略同，我就是这么做的：文档、单元测试，都写成Ticket！

5

邢爱明
2019-04-04

对于详细设计文档的颗粒度一直有点疑问。
是写到类图或者时序图这种级别，说明不同类和方法之间的关系？
还是要细化到类似于伪代码级别，需要写操作哪个数据库表，和调用哪个api接口。这种写法比较费时间，但写出来后写代码就很容易了，基本上是按照开发语言的语法要求进行转换。我在项目中试着曾经推广过，但项目组成员认为工作量太大，基本是应付了事。
希望老师能根据自己的项目经验，看看写到哪个程度比较合适？

作者回复: 我们2002年学软件工程的时候，推荐的写设计文档就是你说的这种细化到为伪代码级别，当时初衷是学习建筑行业，把写代码变成像搬砖砌墙一样，招一堆蓝翔培训出来就可以写代码。据说当年日本软件产业就是这样的。

实际上这些年下来，这种方法是不可行的（至少我没看到过成功案例），一个是设计文档写得太细，其实成本上已经跟写代码没差别了，不利于分工协作；另一个是写代码本身是一种创造性的劳动，当你把文档写到伪代码那么细，具体负责代码实现的没什么好发挥的空间了，都变成体力劳动了。

推荐的做法是写设计文档时不要太细，同时应该把具体模块的设计交给负责这个模块开发的人去做，指导他完成设计。这样既可以更好的分工协作，也可以让程序员有机会成长和充分发挥其主观能动性。

共 2 条评论 > 4

bearlu
2019-04-02

谢谢老师，上一期，按你提示，写了一个自动化静态代码检查工具，省了好多时间，现在学习机器学习也按项目来划分，将机器学习的每个知识点放到看板上。每完成一个就拿掉一个，感觉好很多。

作者回复: 👍👍
很高兴看到你能学以致用，并且有效果！

3

LiYanbin
2019-09-18

宝玉老师，您好，最近在帮忙项目组组员评审他们的模块设计文档。但是怎么帮他们评审却无从下手，不知道评审的标准是什么，好的文档应该要具备哪些点，宝玉老师可否给些看法

作者回复: 我觉得写好技术文档确实不容易，但也没有那么难，毕竟都是给自己内部看的，不要求特别好。

在评审技术文档或者你自己写技术文档，你可以从几个角度去思考：

1. 文档是否讲清楚了它的目的是什么？内容是否和目的匹配？
比如说你这个设计文档是要解决什么问题？设计的是哪个模块？

2. 文档是否解释了为什么要这么做？
比如说你这个模块设计文档，是否解释清楚了为什么要这样设计？这样设计的优缺点是什么？

3. 文档是否描述清楚了如何实现？

比如说作为模块设计文档，到底是如何设计的？有没有讲清楚整体的设计架构？

总结一下就是一个技术文档，要讲清楚：是什么（What）？为什么（Why）？怎么做（How）？这三个是最核心的要素。

这三个核心问题讲清楚了，然后就是多画图，内容简洁明了。

共 2 条评论 >

2

阿G聊产品

2019-04-09

老师，需求分析之后是不是应该还有产品需求分析文档（PRD）和产品需求规格说明书？一直不清楚项目中各个阶段应该具备的文档，所以很困惑。

作者回复: 其实不必困惑这个问题，因为这本身没有特别的标准的。如果用瀑布模型开发，确实会有你说的文档，但如果是敏捷开发，可能会是另外的形式存在，例如每个小功能一个独立的用户故事，或者是独立的产品设计文档，只是讲清楚一个功能。

虽然形式不一样，但其目的都是一样：让大家可以讨论需求，可以理解需求。

之前我在另一条有回复过有哪些文档的问题：

1. 这件事需要讨论需要评审，要有文档作为讨论的依据，以及记录讨论的结果。比如各种设计文档
2. 这件事要有规范，要有文档保证规范统一。比如各种规范文档
3. 这件事要记录下来，作为以后的一个参考。比如各种报告、环境配置、操作手册、API文档等

2

javaadu

2019-04-06

我们组的文档处于中等水平，对外的接入文档很详细，对内的新人上手文档略有欠缺。看到老师说的那个到新团队后的习惯，我仿佛看到了知音，因为我最近就是这么做的。

看过一本书—数据文明，里面有个观点很好：对数据的记录的精细和全面的程度可以反映出文明的程度。放在软件工程的文档这件事上也很有指导意义—文档的清晰和全面程度反映了团队对项目的控制力。

我写文档重度依赖几个工具，推荐给大伙：语雀，印象笔记，xmind，omnigraffie，xnip截图，ppt

作者回复: 谢谢分享!

我也是mindnode（有时候配合xmind），omnigraffie的重度用户。

2

Bo

2019-04-04

谢谢老师哈，特别感谢🙏

作者回复: 不客气，有问题欢迎留言。

2

williamcai

2019-04-03

项目组很少写文档，除非公司合规部要文档，就会冲冲的来一份，但是要回溯以前的内容，很费事。现在也没有人牵头弄这个事情，所以还是流程不规范，是以后的一个隐患

作者回复: 可以尝试把一些关键的必要的文档，作为开发任务的一部分，分配时间，应该会好一点。

2

ownraul

2019-04-03

在开发过程中，比较习惯将一些重要的point归纳后罗列出来，因为有时很简单一句描述需要的代码并不少，过一段时间从代码中反推逻辑还是需要花些时间，不如直接写出来，或者的方法就是直接抽到一个方法中，让方法名来说明

作者回复: 赞，谢谢补充分享👍

2

Geek_long

2019-04-02

深有体会，没有文档的项目，对于项目的维护和交接代价太大了。

2

陈珙
2019-04-02

我偶尔会开源一些自己的写的小项目到GitHub，因此会使用Markdown写read me.md。开始写之前得过一遍基础语法，然后可以参考一些成熟的开源项目的readme，最好找与你的需要写的相似格式的。如果你不熟悉，又是从零编写，写的时候可以先写内容，后套格式语法，这样可以尽量保证写作的连贯性。其实markdown不难，基本上可以现学现卖，很低的学习成本，比学一门新的计算机的语言容易多了。

作者回复: 是的，Markdown很简单的，而且学会了特别实用!


2

小老鼠
2019-09-12

1、就像测试代码一样，文档更新管理有什么好的建议，多久更新一次文档？2、如何理解代码就是最好的文档，代码可以替代文档吗？3、如何甄别无意义的文档?

作者回复: 文档的更新通常是两种方式结合:
1. 即时更新，在内容发生变化时即时更新，或者发现文档不正确直接更新
2. 定期更新，根据团队的特点，定期例如每个月对关键文档进行审查和更新，至于周期设为多久，取决于你的团队能容忍多久不更新的文档

文档的更新要尽可能简单方便，不宜过于繁琐，当更新文档的成本过高，就会越发的没有更新文档的动力。

代码不能替代文档，好的代码可以帮助你理解代码。代码的问题在于很难直接通过代码看到全局架构，必须要配合有文档来从整体说明整体的架构。

为什么有人说代码就是最好的文档呢？并不是代码能替代文档，而是说阅读好的代码，能帮助你快速的理解代码的含义，不需要额外有文档说明；还有就是一些类似于繁琐的一步步的操作部署文档，完全可以用自动化脚本代码替代，从而不需要文档或只需要简单的文档。

没有价值的文档即无意义的文档!


1

老张
2019-04-16

写文档是一个梳理过程!
说出来的东西逻辑性不足 错误不少，写文档的过程是一个优化、排错、提炼的过程，是思维由混乱变为条理的过程，不可或缺。
非常赞同文中的方法:由脑图而ppt,再到文档分节，然后补充内容。

作者回复: 🧡是的，写文档从粗到细比较好些

1

阿G聊产品
2019-04-09

老师，我觉得一个项目中，文档应该分阶段、分角色；比如在项目开始时，项目经理应该写什么文档、产品经理应该写什么文档、开发应该写什么文档等等。但是对于这个我们公司做的很不规范、很模糊，一般都是项目经理兼职产品经理，甚至开发兼职项目、产品的角色。所以您这边能帮我们总结一下吗?

作者回复: 可以参考我给hua168留言的回复和给你另一条留言的回复。

1

一路向北
2019-04-02

我们的项目文档基本上是以协议和流程为主，写这类文档的时候，实际上已经把项目的每一个细节都考虑清楚了，经过几次的review之后，后面的项目实现就是根据文档的内容再继续细化，一旦遇到不太清楚的地方，再回头翻阅文档，也很容易知道当初设计的时候是怎么一回事。
套用格式确实是一个比较好的方式，填空总是比直接写作文要简单的多，而一旦整个空都填满之后，再继续润色，细化那又会比一开始简单些。
写文档，记笔记等，用对工具还是很重要的。

作者回复: 👍有价值的总结分享!

1