

博客分析

1. 第9点

- 内容如下：

对于外国人来说，中国人并不容易合作，主要原因是外语水平。中国的内部市场很大，几乎所有项目都有自己的中文服务，因此中国人接触英语的机会很有限，如果要跟他们清晰的沟通可能是一件很棘手的事情。

这意味着，你几乎总是与实际从事开发的程序员，至少隔了一层。通常会有一个具有一定英语技能的客户经理，你与他联系，他再把你的意思翻译给技术人员。这样的后果就是，当你真正需要进行详细讨论时，没法跟程序员直接沟通。中国程序员会在内部自己商量，你不知道他们在想些什么，并且很多东西在翻译过程中丢失了。

这对代码质量有深刻影响。从功能上讲，代码本身可能还不错，但这是很少或几乎没有英语技能的人编写的代码。高质量软件的要求之一，就是代码本身就是自我记录的文档，这一点很难实现。他们的代码可能无法清晰地描述软件的功能，不仅对作者也对将来维护的人，都很难阅读。

要是你愿意一直跟同一个软件供应商合作，那可能不会成为问题，但如果你收回代码或交给其他人维护，就会遇到严重问题。我的切身体会是，很难评审中国程序员编写的代码。另外，语言障碍对用户界面也有影响，那些编写代码的人由于不太了解英语，编写的标签和标题可能会使英语用户不适应，这意味着要进行许多意想不到的（且预算未定）的修订。

- 内容分析：

上述三段重点在于两个字——沟通。涉及两类人的沟通，一是外包程序员与需求提供者之间的沟通，二是程序员间通过代码形式的沟通。先一次述说上述两点。

I. 程序员与需求提供方。这边的沟通是第一步，准确的提取出用户需求，是软件成功正确开发的前提。如何提取，交流是必须的，那么必须得克服语言障碍，克服语言障碍，如果选择第三者，那么交流必然有嫌隙，双方都不能知道对方想的是什么。那么有何正解呢？最直接的办法就是程序员学好英语。这问题又来了，程序员就是不会英语咋办呢？可以用一些直观的方式，也就是大家都能明白对方想的是啥的关系，例如图形化的方式。不过，最好的方式还是程序员懂英语，与需求方进行直接的沟通，获得需求，进行建模，然后开发软件。

II. 程序员与程序员。这主要是通过第三方作品来进行沟通，第三方作品即代码。编准规范统一的代码风格，合适不多余的注释，以提高代码的可阅读性，进而提高代码的可维护性。这其中需要尤为注意的就是变量和函数的命名，一定要符合规范，能够诠释出它自己的含义，这相当有利于其它的开发者阅读你的代码，修改你的代码，同样符合规范的代码。（大部分）代码本来就是英语，诠释出含义的名字，能够充分解决代码阅读、英语用户等诸多问题。

- 总结：

感慨就是学好英语，这不仅能充分的帮助与外国人沟通，充分理解需求，还能够命名出自解释的变量名和函数名。

2. 第17点

- 内容如下：

第三点，如果想让外包成功，最好采用混合模式。不是“将所有事情外包”，而是“让我们的人与他们的人一起工作，各自做自己最擅长的事情”。

避免依赖外包组织中的个人，比如上文的 Avni，外包行业的人员流动性比一般情况大得多。相反地，应将重点放在让更多人一起参与，如果其中一个离开，你就不会损失太多的项目知识。

还可以多应用一些开发工具，比如代码质量检查工具、构建服务、版本管理等等，让工作过程变得更加可预测。

- 内容分析

这是外包成功的经验，这三段共讲了三点：混合模式、不要依赖某个人和多用开发工具。现依次讲解一下自己的理解。

I. 混合模式。完全交由外包公司负责，那么除了在交接项目时，能对项目的情况有个了解外，之前，对项目是一点了解也没有，例如，项目具体遇到了哪些难点，解决了多久，怎样解决的，对项目的影响是什么。将自己的人也安排进入外包公司，能够有效监督项目走向、项目进展、项目详细情况，最终能够得到一个符合自己公司要求的软件项目。

II. 不依赖某个人。这个可以从两方面理解，一是那个人中途发生意外，包括但不限于生病、跳槽、车祸，发生这些事，将导致项目的停滞和不必要的损失。二是人非圣人，谁能无错，如果依赖的那个人不小心犯了错误这将造成项目不可预估的损失，如果很多人，还能相互之间指出，犯错的概率会大大降低。

III. 工具的使用。很多软件的过程靠人来记录几乎是不可能的，人会犯错，还会遗忘。这个时候就需要充分利用工具，对软件过程进行管理，例如版本管理工具，能帮助人们完完全全的记录整个软件的变化过程，再例如自动化部署工具，能使得开发的项目，很容易的部署。

- 总结

这三点都是成功的经验，是对自己今后有帮助的，今后自己不一定外包，但多多少少都会涉及到软件的开发，那么就应该尤其注意上面的两点，一是不要依赖某个人，二是多使用工具。