

写这份儿文档，我数次压制住了自己的冲动，希望今天写出来的，更多的是一些成熟的思考和建议。

坦白的讲，入职华为还不满一个月，刚刚承担起开发的任务，对公司、部门的认识难免偏颇，此时进“万言书”实在是难避夸夸其谈之嫌。但已然答应了您的要求，我就从我接触到的角落，写一写一个互联网“老兵”眼中，华为第一印象，更准确的讲，在 OC 组这一个月的点滴思考。

第一个我想谈谈的是沟通问题。我们的设计文档粒度实在是太粗了，更像功能模块简单划分，也没有每个模块的设计描述文档。很难准确传达设计者的意图。每个非功能 leader 开发者，更难以捕获设计者的意图。我们以前是这么做的：系统设计师设计出初稿，在完善设计稿的过程中，多次召集每个模块开发团队 leader 讨论，这样开发组的意见能够从早期就进入设计者的考虑，开发组也能从全局明白每个模块的职责，性能压力要求等，这更有利于模块设计。每个开发组中，组 leader 出草稿，带领全组员一起完善设计稿。每个具体的开发人员都对整个模块都十分清楚。同时每个开发人员都出自己的功能设计文档（非常细的粒度），介绍自己具体开发计划和相互沟通。这样，每个小组成员几乎都可以相互 review 彼此的实现代码，代码质量也更有保障。还有一点，代码依赖 maven 仓库加以管理使用。问题是，我们的仓库里面，竟然没有每个 module 的源代码。在设计文档，API 文档都缺失的现状下，大家只能通过反编译来了解逻辑，这显然给每个人都带来了不必要的不便和效率降低。其实解决这个问题非常简单，每个人在往 maven 仓库发布代码的时候，配置一下（maven-source-plugin），source code 和 bin 都发布到 maven 仓库。别人使用这个组件的时候，IDE 就能自动关联上格式良好的源码。

再说说开发流程。开发进度压力可能是太大了，用晓君的话，我们这是裸奔模式。我认为十分必要的流程，比如 code review，单元测试等都很大程度上省略了。这无疑，会降低代码的质量和效率。比如代码普遍没有注释、代码拼写错误泛滥、类设计不够成熟（比如职责不够清晰、很多地方重复造轮子、很多工具方法，已经依赖的库都已经提供，非常优秀的库，e.g. Google 的 guava 不用，反而自己浪费时间去写，代码执行效率上还没人家的好）。再说一点，目前所有人都在 trunk 分支上开发...其实，在互联网公司，基本都是 feather-per-branch 的开发模式，这种模式很成熟，用的公司非常多。见

<http://nvie.com/posts/a-successful-git-branching-model/> 和
<https://www.atlassian.com/git/workflows>

上面都讲得非常具体的小问题，下面说点儿宽泛一些的。坦诚的讲，华为没有工程师文化。每个开发者，几乎都完全是专注在自己的工作任务上。没太多时间去学习、切磋技术。同事间的沟通，几乎完全是在沟通业务流程。当然，专注工作很好，不过及时更新自己的技术储备，可以更好的提高自己的效率。举个例子，JDK7 已经发布两年了，JDK8 都已经发布好几个月了，我们还在使用 JDK6。要知道 7、8 带来了很多提高效率的功能、语法、执行效率，我们都没有或者没办法用来加速我们的开发进度。几乎每个互联网公司（最著名的当然是 Facebook）都有自己的 Hackathon 活动，这样的活动，既可以开发很 cool 的东西，也可以

给产品开发平时没有时间去做的 feature， 总之，有机会践行一下极客的精神。

一点期待：听说公司刚刚允许申请双显示器了， 这个真的应该早点推广，在文档和文档代码之间来回切换，真的很影响效率，尤其是瘦客户端性能配置这么低。顺便说一句，前东家给开发人员用的是 Macbook Pro 再外接 27 英寸 Mac 显示器。

改变，正在发生

鉴于我们的代码库里面有太多的拼写错误，开源界也没有很好的代码拼写检查工具，目前利用业余时间，开发了一个 maven 的插件 `spell-check-maven-plugin`，希望将来能够在往仓库提交模块之前，智能给开发者提示拼写错误。不过目前还不是很成熟，尚无法直接应用的项目管理之中，求 partner 中...

我们的产品有很多南向、北向 REST API 接口，很多连准确的 word 文档都没有，这给 REST 接口的消费者很不方便。有个开源的项目 Swagger <https://github.com/wordnik/swagger-ui> 正好能解决这个问题，让开发者在写代码的时候，像 Javadoc 一样，生成漂亮的、有交互能力 API 文档，参考例子：<http://petstore.swagger.wordnik.com/>，这样，小组之间，能够及时、准确的了解相互的 REST API。

目前的 OC 主要是把下层产品的信息抽取出来，做一个集中的运维展示。对数据的挖掘非常少，这很难形成技术壁垒和持久的竞争力。目前，我和我师父（张赛）还有郝磊同学，正在学习斯坦福大学机器学习大牛 Andrew Ng 的公开课 Machine Learning <http://v.163.com/special/opencourse/machinelearning.html>，希望 OC 将来的版本中，能利用机器学习，做出一两个特色的功能。