

邯郸软件开发-系统开发-网站开发服务-创客情

产品名称	邯郸软件开发-系统开发-网站开发服务-创客情
公司名称	河北创客情创业服务有限公司
价格	15000.00/套
规格参数	
公司地址	河北省邯郸市邯山区光明南大街城市新秀商务大厦18层
联系电话	0310-3334555 18303030203

产品详情

1.系统流程梳理

以一个很简单的例子来说明流程梳理对软件开发的意义，比如你要进行一次演讲，但是这次演讲是即兴的，你不是的即兴演讲家，那么在沒有准备情况下，你要对着台下的人进行演讲，这个时候你走上台去，脑子里的东西还没有形成有条理的演讲内容，讲完后台下的人都不知道你在讲什么，可能你自己都不知道你刚刚讲了些什么，这就是失败的演讲，没有做好充足的准备。对于软件开发来说也是同样的情况，每一个开发者不应该仅仅拿到的是一些文档，而是应该大家坐在一起，由熟悉该软件业务的管理者或者其他人来进行一次严谨的描述，并进行讨论，加以完善和改进，让参与编码的开发者在这个过程中不仅能够熟悉自己要做的那些功能的细节，还能对这个系统有一个大致的了解和熟悉，只有这样，在开发中才会避免一些不必要的问题发生，而且还能发现一些隐藏的问题，要知道修改问题是需要花费很多时间和精力，比如编码和业务是有冲突的，本人有遇到过，代码不能完全跟着业务走，业务也在适当的时候在满足正常场景下根据编码风格做适当的调整。终达到一种整体和谐的一种美感。在编码的前期要让每一个参与项目的人能够清晰的知道我要做的是什，终的目标是

什么样的，我要关注的重点有些，还有哪些疑虑我需要讨论或者解决的。准备工作做好后，对每一个团队成员项目的进度是非常清晰的。

2.技术框架的选择

一般选择技术架构有几个衡量的点：

点：效率。

在开源领域能完成同一个技术目标的框架是多个的，比如在web开发的，最终开发出来的产品是要经过性能这一关的，如果选择有误，整个软件可以说是失败的，因为不能用，你需要重新选择技术框架，并且要重新让每一个开发者在新的框架上进行开发，这是在开发一个新的软件。

第二点：成本。

一个是学习成本，第二个是经济成本，只讨论学习成本，因为本人非常反对使用商业软件，把这笔买商用软件的资金来激励和培养员工效果会更好，这里不做什么讨论，不是商业上的东西就很安全，开源的东西也很安全，只说一句：大部分情况都是浪费！关于学习成本要考虑到团队实力和团队人才培养方式，如果项目团队没有什么培训和学习气氛，那么这个团队选择框架的原则是非常简单的，在这种情况下就选择自己熟悉的能有把握的；还有一种情况就是团队中有实力非常强的开发者或者学习能力非常强的开发者，那么可以选择一款相对适合整体架构的新技术框架，并加以重视，因为这是新的东西，风险也是非常高的，只要重视了，而且技术上可行的，结果是完美的；这是根据团队的实际情况进行参考，勇气也很重要。

第三点：稳定性。

选择一个合适的软件版本，个人比较倾向于在新的平台和框架上进行

开发，因为有新的特性，有可能心的版本有进行一些优化。

对稳定性的考虑，举一个例子，根据实际情况已经选择要使用一个A框架了，假设A框架有两个版本，V1和V2，V1是稳定版本，V2还是测试版本，V2中添加了一些新的功能，而这些功能正好满足你的项目需要，并且稳定版本是在你编码完成前就会发布，那么眼前有两个选择个选择，选择V1版本并且要选择一个新的B框架来满足项目需要，这种方式风险是低的；第二种选择，选择V2测试版本，终等到稳定版本发布后进行替换，这种方式也是可以选择的，不过风险相对种选择要高些，有一个优势就是这一个框架就可以完全满足你的项目需求，成本相对低一点。两种情况我都有实践过。

3.编码

在软件产品的编码中需要注意的一些宏观问题：

点：代码风格。

一个年轻的团队很容易遇到这个问题，一个软件开发完了，回头去看里面的代码，编码风格很不统一，有5个开发者就有5种代码风格！怎么样避免这种情况，只能在编码之前进行代码编码风格宣讲和讨论，把规则制定下来，大家按这种风格进行代码编写，还有一点要做的就是代码检视，不要因为忙而忽略这个，一周花一个下午来看看别人的代码，不仅能看到一些问题，而且还能看到自己的一些问题，当开发一段时间过去以后，代码不断的调整，终的源码看上去就是一个人完成的一样！所以开工之前把这方面工作做好，事半功倍，后面还有很长的软件维护工作要做，如果整体代码一团糟，我想没人愿意去维护这么糟糕的代码。这样的项目本人也遇到过，深有体会。

第二点：注释。

比风格统一的更难的可能就是注释了，我想你不会这么认为，我也想自己这种认识是错的，因为写注释这种活总比编码要容易得多吧，不是这样的，很多人应该都看过国内一些开源的程序员写的开源软件吧

，很膜拜吧？呵呵，我也有看过，说下我的感受吧，首先代码很少有注释，一个类文件看下来只有代码，注释非常稀少，不知道他是怎么想的，再简单的代码也要有方法和类注释吧；其次，代码里面有稀疏的注释，好不容易啊，结果是英文的，还有文档里面都是英文的，一个说中文的家伙为什么搞成英文版的呢。另外，打印日志不加级别判断，还有一些编码问题在里面。很想骂几句，但是人家毕竟是开源的，不容易啊！精神可以鼓励，但是态度值得怀疑。如果你现在刚编完代码或者要开始编码了，请把代码写好的同时把注释写好吧！如果一个刚入门的程序员能直接通过注释就能读懂你的程序代码，那么你写的注释已经非常成功了。

第三点：代码目录结构。

这点和编码风格是挂钩的，也可以属于代码风格里面的一部分，但是单独拿出来肯定有独特的含义。你有没有想过或者遇到过通过代码目录结构就能够大致看懂该项目是要做什么，有哪些功能，如果看到这样的工程是不是有一种很想再往里面看的冲动？本人有参与这样的项目编码，当时我们做的还比较成功，刚开始做有点不习惯和编码风格不同，关于代码目录结构我们进行了单独的讨论，根据本身的技术架构来制定的，把这点做好，开发者编写代码更加清晰了，效率也有所提高了，后期维护哪怕是新人来维护，只要稍微讲讲，也会很容易的接受，一切都变得更加简单了。

第四点：命名。

这点也可以同属于代码风格。坦白讲单独拎出来说也没有多大意思，因为代码风格里面就会强调，但是你不觉得这么重要的东西很容易忽略吗，比如大小写，id我是写Id还是写成ID呢，没有多少人会在意，只有出现问题了，代码冗余量增加了，才会发现，命名也是非常重要。还有一些，类文件的命名词不达意的，我想提醒你的是，既然这么重要那么请谨慎对你的代码进行命名！

第五点：赞成有必要的重构。

重构需要注意时机，有两个点是好进行重构了，点是在自己编写完代码以后进行优化和重构，转测试之前；第二点就是当项目初期大家没有意识到要去重构，也就是点没有做充分，导致代码重复率比较高等一些整体问题，在这种前提下找一个时间段，对整体代码进行一次重构计划，这是有必要的。

第六点：一些提高代码的工具使用。

在这里简单列出几类工具，网上有很多资料，需要根据自己的语言进行选择。

类：代码自动检视bug工具

第二类：代码统计工具

第三类：代码重复率和复杂度工具

第四类：代码覆盖率工具

第七点：不要随意修改代码，特别是别人的代码。

修改代码应该是放在一个时间段，而不是随意进行修改，目前比较流行的敏捷开发中有一个现象就是版本发布比较频繁，修改代码是有很大的风险的，修改的代码很有可能是公共代码，多处地方有调用，很有可能造成其他地方出问题，小问题解决，大问题来了。当需要修改其他开发人员的代码时一定要和对方沟通下，避免造成不必要的误会和引发潜在的问题。

*编码中需要注意的一些微观问题

这些就是编码功底了，我自身的感受就是，要不断的回头看看自己的代码，很多地方值得你重新思考和关注。

平时有时间可以静下心来阅读比较经典的书籍，看不懂或者不太记得

没有关系，重新再看。

4.测试

作为一个开发人员所接触到的测试首当其冲的就是编写单元测试用例，尽量覆盖每一个场景，这对软件质量起到一个很关键的作用，为了避免与测试人员反复沟通增加无谓的成本，开发能做的就是写单元测试发现一些潜在的问题，把大部分的bug提前发现。从管理角度来讲，测试也会轻松很多。开发一款相对完美的软件是一个程序员的追求。也是在程序员这条道路上的一笔收获。