目录

[1. 双显示器 1](#_Toc363723401)

[**2.** **双机器** 1](#_Toc363723402)

[3. 抛弃XP,使用64位WIN7系统 2](#_Toc363723403)

[**4.** **打印文档资料** 2](#_Toc363723404)

[**5.** **使用休眠，而不是关机** 2](#_Toc363723405)

[**6.** **使用SSD，抛弃普通硬盘** 2](#_Toc363723406)

[**7.** **工作地点的选择** 2](#_Toc363723407)

[**8.** **工作季节** 3](#_Toc363723408)

[**9.** **提供宿舍,减少路上奔波浪费** 3](#_Toc363723409)

[**10.** **弹性工作制，补休长假** 3](#_Toc363723410)

[**11.** **集中式开发** 4](#_Toc363723411)

[12. 扔掉MVC，使用WINFORM技术(或者JSF)。。 4](#_Toc363723412)

[13. 不用SPRING作IOC。。 4](#_Toc363723413)

[**14.** **扔掉HIBRENATE，使用代码生成器生成代码。** 4](#_Toc363723414)

[15. 只要较大模块间使用INTERFACE。。。 4](#_Toc363723415)

[16. 使用双层架构 5](#_Toc363723416)

[17. 参考 5](#_Toc363723417)

# 双显示器

给开发电脑配置双显示器，可以极大拓展工作平台界面。。

并且可以大大减少窗口切换

1. **双机器**

开发者最好拥有多台电脑，可以减少窗口切换，以及性能挥发。.

要是机器性能比较差..使用这个方法相当于多核了…

# 抛弃XP,使用64位WIN7系统

64位系统能使用大内存….一般来说,4g内存应该够了..要是能达到6G就越好了..

1. **打印文档资料**

将相应的文档打印出来，随时翻阅，也可以大大减少窗口切换..

1. **使用休眠，而不是关机**

电脑应该设定为休眠状态。。这个重启动后速度非常的快…且工作状态不会丢失

1. **使用SSD，抛弃普通硬盘**

SSD的速度更快，开机速度立马击败全国95%的电脑…

1. **工作地点的选择**

最好是环境优美的地方，海边，湖边,森林边是工作的良好环境，。。使人的工作效率大增..

1. **工作季节**

对于软件开发来说，冬季是个是最提升效率的季节。。因为外面冷，哪里也去不了。。呆在电脑旁边会更温暖。呵。。所以工作地点如果可以选择到南北极就最好了。呵…

1. **提供宿舍,减少路上奔波浪费**

每天上下班两三小时，完全是种大大的浪费，把这些宝贵的时间放在软件开发启不是更好。。另一方面，上下班拥挤让人精神不爽，损失精力…

1. **弹性工作制，补休长假**

比如在一个月的时候内不放假，全月30天全力以赴。。然后补休个10天长假，顺便可以旅游，放松身心。。。。取消固定的5天工作制。

..工作时间可以一天12小时制,可以多发加班工资来平衡,再进行一次长补休。。。

这样一个月全力以赴30天\*12小时，然后多发加班工资，一个月后直接补休一个月..工作一个月然后放一个月假.呵呵。。。

1. **集中式开发**

将异地人员集中在一地，开发，减少沟通环节的损失…

1. 扔掉MVC，使用WINFORM技术(或者JSF)。。

事件驱动。更快，更好理解

1. 不用SPRING作IOC。。

非常影响开发效率 以及 运行效率..

一般的项目就不必使用IOC,以及spring

1. 扔掉HIBRENATE，使用代码生成器生成代码。

开发效率执行效率都非常快..

1. 大量使用代码生成器

**代码生成器投入使用后，对我们项目开发的资源投入的改善是非常明细的，目前基于基础平台和代码生成器的配合，我们已经做了6、7个系统了，平均每个系统的开发时间至少要比以前节约40％，有的项目甚至达到了80％以上（我们最高的一天，处理了40多个表的增、删、该、查的功能，及中文本地化）。而且，另外很重要的一点，生成的代码无形中统一了程序员的设计风格，我们通过这套开发机制，能够最大限度的保证我们开发的系统质量，保证模块可以在不同系统之间的自由迁移，最大限度的实现复用！在项目开发中节省出来的大量时间，也让我们可以去研究更多的开源中间件和系统，来增强我们的基础平台，从而形成一个良性的循环！**

# 只要较大模块间使用INTERFACE。。。

目前太多的项目在模块内部功能间都使用INTERFACE，设计过度了....

模块之间，需要接口,重要元件之间需要.

# 使用双层架构

一般而言，三层架构过度了。。90%的模块一般双层架构就可以了..极少量模块(10%)使用三层架构

# 新模块立及上线法

只要一开发一个新模块立及上线使用.这样一可以开发速度加快,一可以边使用中测试.

外围功能可用其它方式实现, 不必非等到编码..如电话,EMAL,呼叫中心

比如,建立一个CRM.

# 重要功能先上线

一般来说

# 扁平化设计(业务方法,通用方法)

# 简单文档

# 流程标准化

# 组件化,构件化.

# 需求文档>>详细文档,,需求对应多个METHOD..

# 参考

# 使用拼音代替英文

拼音的好处是它跟汉字一一对应.

# 知识管理制度(重要)

包括知识的总结..每个人每天应该要总结多篇心得体会..

知识的检索:根据文件名的检索,根据文件内容的检索

# 异步开发模式，也称并行工程。

就是通过严密的[计划](http://baike.baidu.com/view/149377.htm)、准确的接口设计，把原来的许多后续活动提前进行，这样可以缩短产品上市时间。

异步开发模式的基本思想是将[产品开发](http://baike.baidu.com/view/352523.htm)在纵向分为不同的层次，如技术层、子系统层、平台层等。不同层次工作由不同的团队并行地异步开发完成，从而减少下层对上层工作的制约，每个层次都直接面向市场。

通常，在[产品开发](http://baike.baidu.com/view/352523.htm)过程中，由于上层技术或系统通常依赖于下层的技术，因此，开发层次之间的工作具有相互依赖性，如果一个层次的工作延迟了，将会造成整个时间的延长，这是导致产品开发延误的主要原因。 通过减弱各开发层次间的依赖关系，可以实现所有层次任务的异步开发。

为了实现异步开发，建立可重用的共用基础模块是非常重要的。

# 脚本化

# 提升效率-几款任务栏软件vc59

1. **内部承包制,抛弃分配制**

划分为详细的功能点后，将功能点承包给项目组组员，谁完成的多，薪水就多,绩效挂钩….

功能点可按照不同的难度与时间，设定不同的绩效点..

分配任务的时候，采用抽签制..可最大程序的避免任务分配的人为因素。。然后组员之间可以互相交换功能点.

# 适当外包

# 更改财务结算模式---按件结算

# 周薪制

# IPD框架

# CBB重用性。

# 采用公用构建模块（CBB：Common Building Block）提高[产品开发](http://baike.baidu.com/view/352523.htm)的[效率](http://baike.baidu.com/view/47610.htm)。

# 共用基础模块（Common Building Blocks, CBB）指那些可以在不同产品、系统之间共用的零部件、模块、技术及其他相关的设计成果。由于部门之间共享已有成果的程度很低，随着产品种类的不断增长，零部件、支持系统、供应商也在持续增长，这将导致一系列问题。 事实上，不同产品、系统之间，存在许多可以共用的零部件、模块和技术，如果产品在开发中尽可能多地采用了这些成熟的共用基础模块和技术，无疑这一产品的质量、进度和成本会得到很好的控制和保证，[产品开发](http://baike.baidu.com/view/352523.htm)中的技术风险也将大为降低。 因此，通过产品重整，建立CBB数据库，实现技术、模块、子系统、零部件在不同产品之间的重用和共享，可以缩短[产品开发](http://baike.baidu.com/view/352523.htm)周期、降低产品成本。 CBB策略的实施需要组织结构和衡量标准的保证。

# 软件驱动库FTP的建设

避免安装软件跟驱动浪费的时间

1. **多多COPY网上的模块与功能代码**

a.       从开源项目选型，当需要哪个模块或者功能时，先从网上源码找一个，然后将此模块剥离出来，拼接到项目里，这样速度大大的加快啊。.也少了很多开发测试的时间....记住不要重复发明轮子，呵呵。。

b.

c.       如果是不同语言开发的，做个接口调用就可以了

许多人都喜欢说，非得要自己的代码才放心。这个是个很大的观念问题..只要抛弃先前的成见，就可以大大提升效率…

# 模块化设计:一个模块一个文件夹

在一个现有开源项目中，我们会发现，当我们需要剥离某一个模块时，往往界面与功能并不是存放在同一个文件夹下的,这使得我们剥离的工作量大增..

特别是WEB项目，不得不从IMG，CSS目录中寻找自己模块需要的资源..

解决方案就是: 模块化设计:一个模块一个文件夹

如果一个功能比较常用，可以一个功能一个文件夹..

**$$$$$使用IDE而不是记事本**

书写HTML你是用IDE吗？？

写JS你是用IDE吗？

写CSS你是用IDE吗？？

调试时你是用IDE或者各种工具吗？优先选择工具，而不是在语句中输出来调试..

HTML,JS,CSS,ajax,SQL都有相应的调试跟踪工具,这些代码的调试与跟踪工具往往人们不大知晓…

**\*\*\*\*\*\*语言平台及架构的选择**

尽可能使用。NET 技术，比起JAVA，PHP会开发效率更快…

如果主机是LINUX,可能只能用JAVA和PHP，优先使用PHP，开发效率快..

如果可以选择CS架构，就不要用WEB BS架构，这样开发效率倍增。.

**3小时连续工作点。**

一般而言，8小时连续工作效率低下。。。一个人一般只能集中精力3小时。。所以每隔三两小时休息一两小时是个良策…

# 界面原型设计工具： UIDesigner 界面HTML--CSS  IDE:dreamweaver，没说的， HTML,CSS调试  ，IE F12工具

# 详细设计可用伪码进行..

# JS IDE工具，Aptana Studio ,我主要用来编写JS。自动提示功能很好.. js调试与跟踪  ，IE F12工具 AJAX抓包跟踪工具:httpwatch7 专业版

EASY COCDE: 非常强大的代码生成器

VS2010 t4模板工具：可生成所有代码,需要先写模板。。一般用它来生成WEB界面代码及利用它生成SQL代码

# 参考:

### [paip.PHP-asp—jsp实现事件机制 WEBFORM式开发](http://blog.csdn.net/attilax/article/details/7979788)

**paip.项目开发效率提升之思索**

[**paip.提升开发效率----JAVA网站**](http://blog.csdn.net/attilax/article/details/7941529)

### [提升效率-几款任务栏软件vc59](http://blog.csdn.net/attilax/article/details/8904022)

### [paip.提升效率-源码生成流程图工具](http://blog.csdn.net/attilax/article/details/7993409)

### [paip.提升效率-文件搜索工具总结](http://blog.csdn.net/attilax/article/details/7865820)

[**paip.提升效率---模块化设计方法V2012.9.15**](http://blog.csdn.net/attilax/article/details/7982142) **后缀法避免**使用多个同样的模块，**SESSION/COOKIE/TABLE冲突**

**IPD开发方法**

**SCRUM开发方法**

### [paip. JAVA快速开发IDE与JSF技术](http://blog.csdn.net/attilax/article/details/8062833)