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99%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....

# 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

# 参考

**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)