个人使用理解：可以监视项目当中书写代码的不规范和不好的代码，监视程序运行情况，如果出现严重的问题，会有相应的对话框提示，和对应的log,安卓在2.3API就加入了该API，官方定义：http://developer.android.com/reference/android/os/StrictMode.html，

        发现有这么好个开发工具为什么不用到自己的项目里面来了，举例，把该工具添加到自己的项目，以安防项目为例：添加步骤也很简单，官网说的：Example code to enable from early in your Application, Activity, or other application component's onCreate() method:也就是放到你程序最先初始化的onCreate() 方法里面，也就是我们自己的Application  里面初始化StrictMode ，如下代码：

**if (DEVELOPER\_MODE) {**

**StrictMode.setThreadPolicy(new** [**StrictMode.ThreadPolicy.Builder**](http://developer.android.com/reference/android/os/StrictMode.ThreadPolicy.Builder.html)**()**

**.detectDiskReads()**

**.detectDiskWrites()**

**.detectNetwork()   // or .detectAll() for all detectable problems**

**.penaltyLog()**

**.build());**

**StrictMode.setVmPolicy(new** [**StrictMode.VmPolicy.Builder**](http://developer.android.com/reference/android/os/StrictMode.VmPolicy.Builder.html)**()**

**.detectLeakedSqlLiteObjects()**

**.detectLeakedClosableObjects()**

**.penaltyLog()**

**.penaltyDeath()**

**.build());**

**}**

**官网解释如下：**public static void setThreadPolicy (StrictMode.ThreadPolicy policy)  
  
Added in API level 9  
Sets the policy for what actions on the current thread should be detected, as well as the penalty if such actions occur.  
  
Internally this sets a thread-local variable which is propagated across cross-process IPC calls, meaning you can catch violations when a system service or another process accesses the disk or network on your behalf.  
  
Parameters  
policy the policy to put into place

个人理解：对API要求是在9以上也就是2.3以上，该策略就是对线程使用的一个检测，在内部，该设置是传播跨跨进程的IPC调用，这意味着你可以捕捉违规行为时，系统服务或其他进程访问代表您的磁盘或网络中的线程局部变量。

setVmPolicy：设置检测虚拟机，

一般就两大类：，一类是关于常用的监控方面的，另外一类是关于VM虚拟机等方面的策略，然后我运行程序，出现如下错误：

****

对应的LOG错误如下：  
12-06 11:43:58.615: D/ImageLoader(16738): Initialize ImageLoader with configuration  
12-06 11:43:58.615: D/StrictMode(16738): StrictMode policy violation; ~duration=23 ms: android.os.StrictMode$StrictModeDiskReadViolation: policy=63 violation=2  
12-06 11:43:58.615: D/StrictMode(16738): at android.os.StrictMode$AndroidBlockGuardPolicy.onReadFromDisk(StrictMode.java:1123)  
12-06 11:43:58.615: D/StrictMode(16738): at android.app.SharedPreferencesImpl.awaitLoadedLocked(SharedPreferencesImpl.java:203)  
12-06 11:43:58.615: D/StrictMode(16738): at android.app.SharedPreferencesImpl.getString(SharedPreferencesImpl.java:223)  
12-06 11:43:58.615: D/StrictMode(16738): at com.cleaderwin.anfang.app.AnFangApplication.initConfig(AnFangApplication.java:121)  
12-06 11:43:58.615: D/StrictMode(16738): at com.cleaderwin.anfang.app.AnFangApplication.onCreate(AnFangApplication.java:67)  
12-06 11:43:58.615: D/StrictMode(16738): at android.app.Instrumentation.callApplicationOnCreate(Instrumentation.java:1007)  
12-06 11:43:58.615: D/StrictMode(16738): at android.app.ActivityThread.handleBindApplication(ActivityThread.java:4444)  
12-06 11:43:58.615: D/StrictMode(16738): at android.app.ActivityThread.access$1300(ActivityThread.java:141)  
12-06 11:43:58.615: D/StrictMode(16738): at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1316)  
12-06 11:43:58.615: D/StrictMode(16738): at android.os.Handler.dispatchMessage(Handler.java:99)  
12-06 11:43:58.615: D/StrictMode(16738): at android.os.Looper.loop(Looper.java:137)  
12-06 11:43:58.615: D/StrictMode(16738): at android.app.ActivityThread.main(ActivityThread.java:5103)  
12-06 11:43:58.615: D/StrictMode(16738): at java.lang.reflect.Method.invokeNative(Native Method)  
12-06 11:43:58.615: D/StrictMode(16738): at java.lang.reflect.Method.invoke(Method.java:525)  
12-06 11:43:58.615: D/StrictMode(16738): at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:737)  
12-06 11:43:58.615: D/StrictMode(16738): at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:553)

所以我觉得这个开发工具，应用策略完全可以放到程序里面来，对程序五毒无害，还绿色开源，为何不用了？