

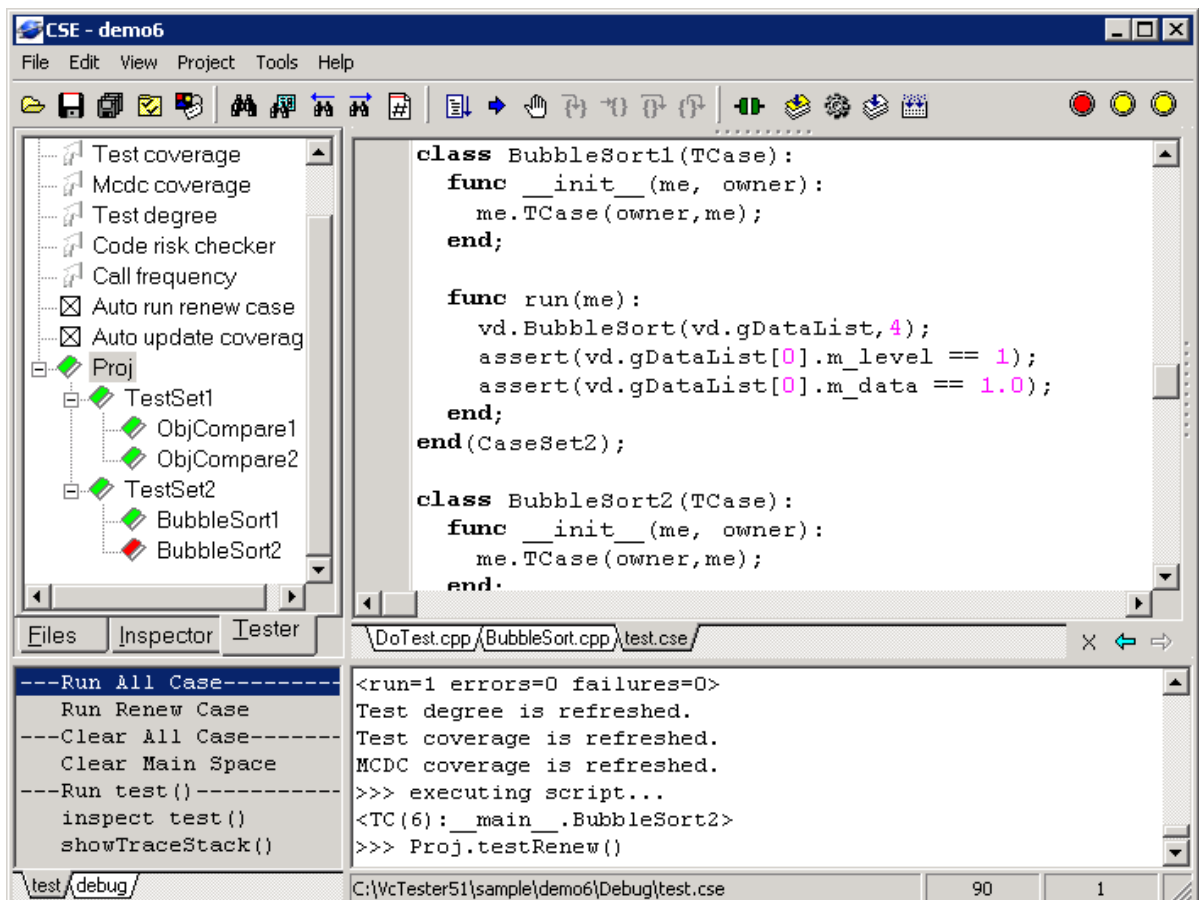


- Ø Visual C/C++编程必备工具
- Ø 高效的覆盖率测试工具
- Ø 完整支持第4代白盒测试方法
- Ø C/C++敏捷开发的最佳载体

VcTester 是与 Visual C/C++ (VC6、VC7、VC8、VC9) 配套使用的专业覆盖率测试工具，它提供脚本化测试驱动、脚本桩、测试覆盖率评估、测试报告生成、被测源码风险检查等功能。

VcTester 率先在 C/C++ 领域完整支持第4代白盒测试方法，遵循该方法论所要求的在线测试、持续测试、调测一体、use case 驱动等理念，让测试过程更加敏捷，测试设计过程也更加高效。敏捷与高效是 VcTester 相对于常规覆盖率工具的最大亮点。

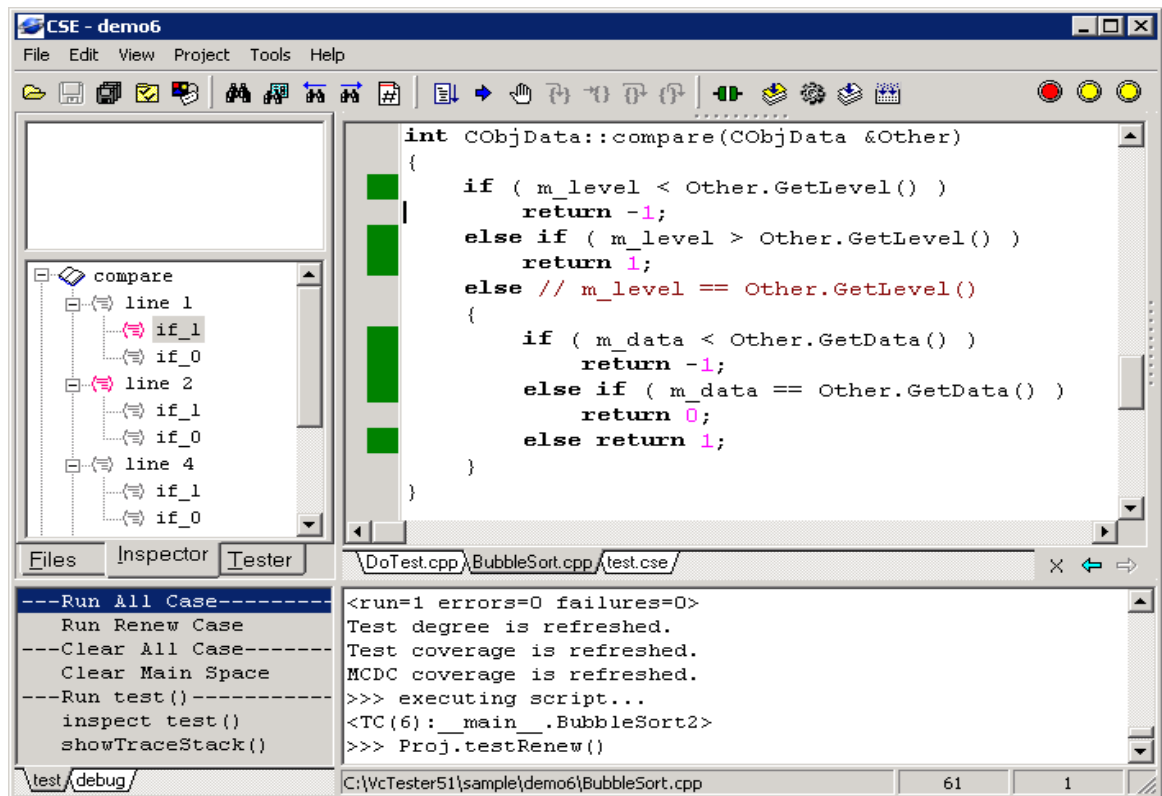
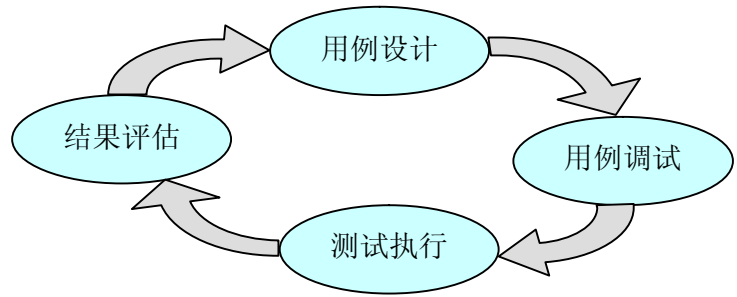
- U VcTester 继承了 VcSmith 强大易用的 IDE 环境。（注：VcSmith 原本是 VcTester 中一个组件，从 V2.2.0 版本开始 VcSmith 既是 VcTester 组件，也同时独立发布版本）
- U VcTester 的敏捷与高效特性源于对 C/C++ 变量与函数实施脚本化映射。用脚本操作 C/C++ 实体从而获得在线脚本驱动、在线脚本桩等特性。
- U VcTester 提供 Mini 版本，让大家免费体验比 CppUnit 更强悍的测试驱动与测试桩机制，请从 [www.ezTester.com](http://www.ezTester.com) 下载最新版本。



## 在线测试

在 VcTester 中启动被测程序后,用 CSE 脚本编写测试用例,选中执行用例,立即可以看到执行结果,用例运行是否通过、哪行语句跑到、哪行没跑到立即可以看到,然后我们有针对性的

完善或补充用例,进入下一轮迭代测试。即:从测试用例设计、用例调试、测试执行、结果评估,然后进入下一轮测试的操作过程都是在线进行的,不必重新编译或重起被测程序。



```
## 定义脚本桩函数: stub_func
func stub_func():
    print(this->m_level,this->m_data);
    print(Other->m_level,Other->m_data);

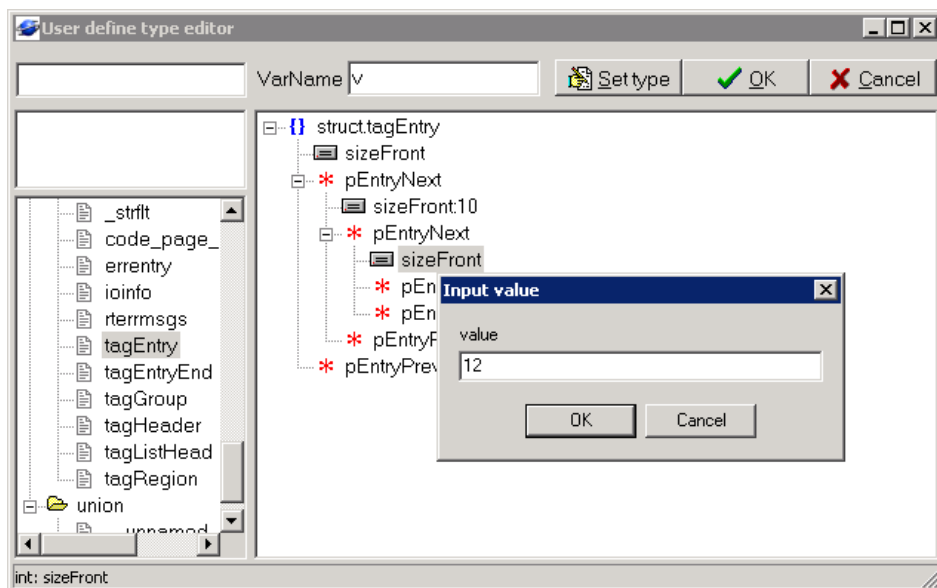
    if this->m_level > Other->m_level:
        return 1;
    end else return -1;
end;

## 打脚本桩,让 stub_func 替代 CObjData::compare
CObjData::compare.stub(stub_func);

vd.DoTest(); ## 发起测试
CObjData::compare.stub(dummy); ## 删除脚本桩
```

## 在线脚本桩

定义一个脚本函数,然后拿这个函数替代被测的 C/C++ 函数,运行中由脚本函数控制测试逻辑,并指定返回值,这就是 VcTester 脚本桩。脚本桩可以即时定义,即时生效,脚本桩中可完整恢复输入参数,即时指定各种返回值以便调用者函数各个分支都能跑到。VcTester 脚本桩除了支持用脚本函数替代被测试函数,还支持将某段测试逻辑插入到被打桩函数之前运行,原有函数不被替换。

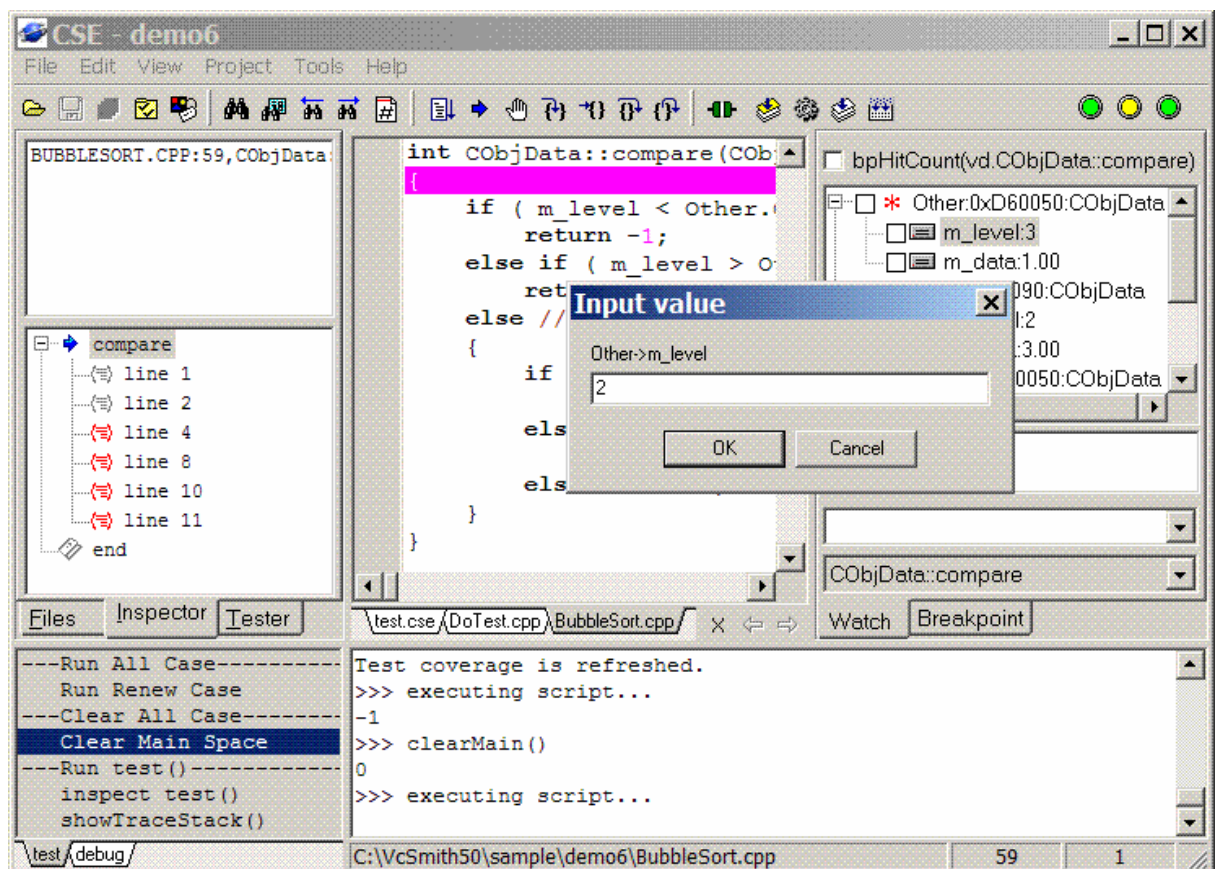


## 高效测试设计

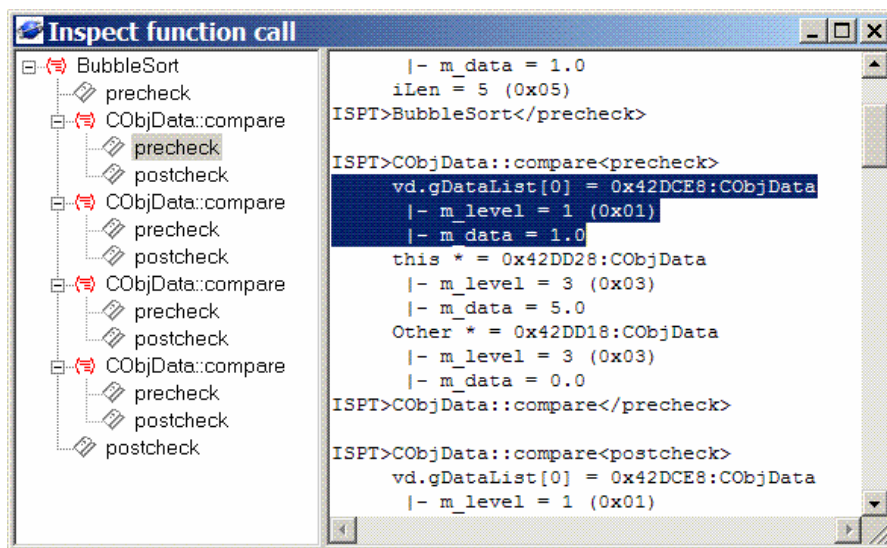
VcTester 提供 UDT 测试数据编辑器，以可视化方式逐步展开测试数据构造，构造过程中支持指针与数组子成员展开，支持指针类型转换，自动生成能自主管理内存的相应脚本语句。

VcTester 支持测试用例脚本框架自动生成，系统自动识别被测函数原型，构造参数与判断返回值的样例语句自动生成。另外，还可从 Watch 列表自动生成预期值判断语句。

VcTester 支持按“录制/回放”的方式设计测试用例，在被测函数中设置断点，然后用脚本发起单步调试，在单步跟入跟出等过程中观察变量是否预期，可修改局部变量或全局变量的值来改变程序运行路径，这些调试操作将录制为测试脚本，稍作整理即成规范测试用例，此功能让测试设计效率倍增。



录制回放借助脚本桩来实现，通过打桩在被测函数入口与出口位置插入一段脚本处理，比如修改变量或参数的值，检查变量或返回值是否预期。经此录制的测试脚本稍作修改，比如修改传入参数为不同取值可构造若干测试运行环境，我们可以快速得到若干个测试用例。



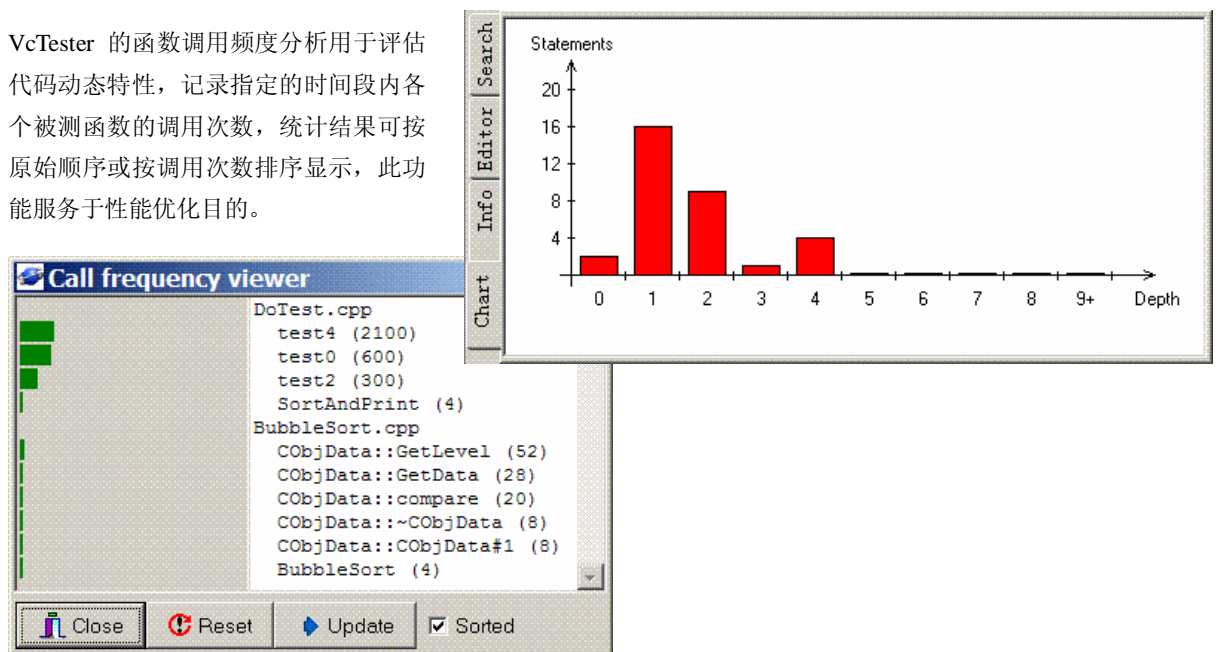
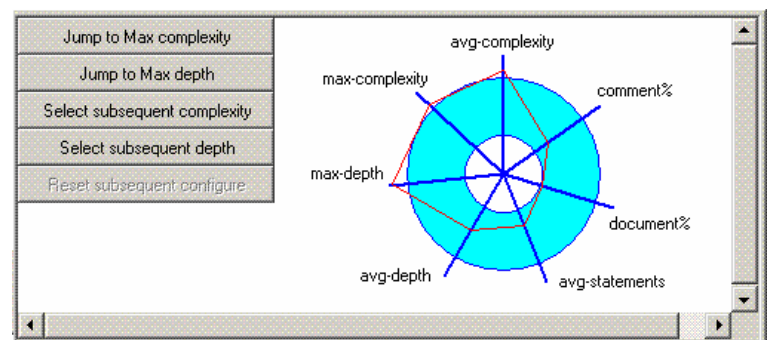
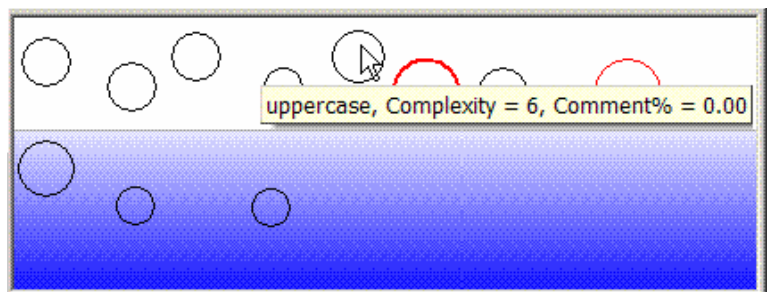
为便于快速发现并定位问题，VcTester 提供函数调用中参数与返回值自动打印的功能。先选择待观察的若干函数，然后用脚本发起调用，在函数调用过程中，各个被观察的函数的传入参数与返回值依次被打印，打印信息以调用树方式直观的显示给用户。除了参数与返回值，全局变量也可以纳入打印。

## 丰富的测试评估系统

VcTester 在代码的静态特性、动态特性，以及测试完整性这 3 个方面提供评估。

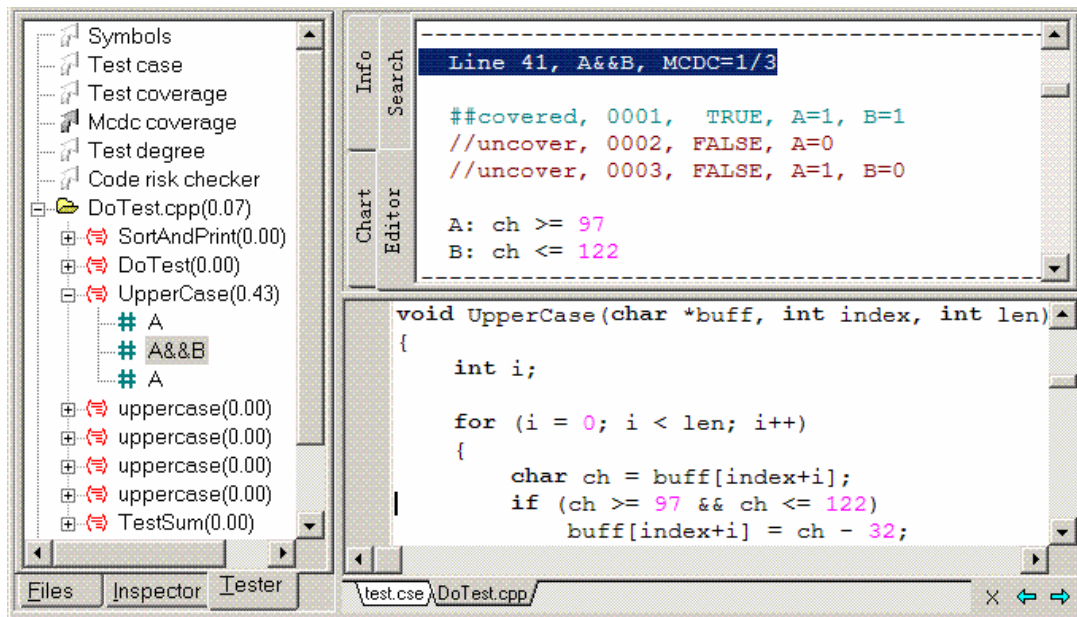
其中，代码风险检查器用于评估被测源码的静态特性，评估项包括代码行数、注释行数、语句数、分支数、函数复杂度、函数调用数等共计 19 项指标，从模块整体到局部函数全面分析被测代码中可能存在的可维护性与可测试性风险。评估结果以冒泡图、直方图、Kiviat 图、统计表等直观的方式表达，给出详细建议以便有针对性作改进。

VcTester 的函数调用频度分析用于评估代码动态特性，记录指定的时间段内各个被测函数的调用次数，统计结果可按原始顺序或按调用次数排序显示，此功能服务于性能优化目的。





VcTester 支持语句覆盖、分支覆盖、条件分支覆盖、MCDC 覆盖的统计分析，除了这些常规代码覆盖率指标，VcTester 还创造性的提供用例覆盖度评估，可以有效克服伪测试导致的覆盖率虚高情况。评估结果既可在线浏览，也能导出为正规测试报告，测试报告采用 XML 格式，适合客户自行定制。



## 从在线映射到离线映射

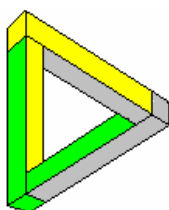
在线映射是在已运行的系统中将 C/C++ 的变量与函数映射为脚本对象，用脚本在线的操作被映射的变量与函数。离线映射是将脚本语句翻译为 C/C++ 语句，编码与调试以脚本语言为主体，当系统测试稳定后再将脚本语言翻译为 C/C++ 语言，从而实现脚本化开发的目的。尽管离线映射不在 VcTester 工具中支持，但离线翻译工具与 VcTester 配合起来使用，可以打造一套高效、实用的敏捷开发实践，让研发进度与产品质量同时得到显著提升。



领测科技有限公司 (ezTester Technology Co., Ltd)

<http://www.ezTester.com>

领测科技专注于复杂软件的 C/C++ 产品测试领域，专业为全球厂商提供一流的白盒测试与自动化测试的解决方案及配套工具。目前，领测科技正致力于推行第 4 代白盒测试方法，藉此促进软件研发质量，提高测试工作效率，帮助客户实现研发产品的市场价值。



CSF 共享软件基金 (CSE Shareware Foundation, CSF)

<http://www.cse-soft.org>

VcSmith 软件由领测科技 (ezTester) 捐献给 CSF，CSE 与 VcSmith 共享用户操作界面，大家可免费享受更多由 CSE 提供用于辅助 C/C++ 编程的功能。