VcTester 持续集成框架的应用价值

2006-10-20

VcTester 是一款针对 Visual C/C++工程中 C 代码的白盒测试工具，它除了支持常规的覆盖率测试之外，还提供功能强大的编辑、调试等功能，该工具遵循第 4 代白盒测试方法，调测理念较为先进。尤其更为难得的是，VcTester 是一个高度开放的 IT 工具，本文着重介绍 VcTester 工具在支撑持续集成开发模式的应用情况，重点讲解 VcTester 共享版在这方面的应用价值。

如果你还不熟悉 VcTester，请到 www.ezTester.com 网站下载最新安装包，根据指示安装 VcTester 后请阅读帮助文档先了解 VcTester 基本功能。

持续集成是极限编程中一项重要实践，采用持续集成开发模式可大幅提高软件生产率，提升产品质量，但目前较成功的持续集成实践主要集中于 java 与 C#项目，在嵌入式领域软件主要用 C 语言开发，开展持续集成的难度要高一些，VcTester 的应用价值很大程度上体现在这里：它克服了编程语言的差异性，让持续集成实践在嵌入式软件开发中也能很好的应用起来。

持续集成对 IT 工具提出的需求

通俗一点讲，持续集成是“代码写一点测一点”的开发模式，“实现—验证—优化”的迭代循环过程加速了，即：为实现功能进行编码，编码后做测试，然后根据测试通过情况，进行改错或优化功能，之后进入下一轮编码与测试，这种递增循环过程在持续集成方式下，迭代更频繁，编码与测试的切换更快速，这对 IT 工具提出了较高要求：

1. 编辑、调试、测试一体化平台

   持续集成模式下，代码写一点点就进入测试的循环迭代了，依据经验，嵌入式 C 语言开发环境下，一次功能迭代编写的代码量宜控制在 20~200 行范围，每写数十行代码调测应随即跟进。这种情况下，IT 工具需支持编辑、调试、测试这 3 项操作之间能快速切换，三者应集成到一个 IDE 环境中。

2. 提高调测效率

   不少商用的第 2 代白盒测试工具对一次性测试支持很好，工作效率能基本满足要求，但一换到持续集成模式下，暴露的问题就很严重，主要是测试效率跟不上，比如在 RTRT 环境下，编码、调试、测试与评估改进既不在同一平台进行，也没支持在线操作，被测程序要反复重起，编写的脚本不能在线的看到执行效果，并在线改进，迭代循环的效率自然大打折扣。

   第 4 代白盒测试方法为提高测试效率提出两点建议，一是采用在线测试，二是重用调试过程，将调试操作自动转化为测试脚本，测试效率会提高不少。

3. 保证持续集成过程能平稳、可持续的推进

   代码写一点测一点后，组织运作很容易失去控制，自觉些的会认真跟进测试设计，不

觉得的可能退回到一次性测试模式，或者某些员也很想把持续集成规范的实施好，但操作中很难控制当前要测到什么程度才进入下一步功能迭代。

IT 工具应提供机制来评估当前测试程度，评估结果应以简单明了的方式告诉用户，比如当前测试是否通过，覆盖率有没有合乎要求，测试充分程度有没有合乎要求。凡测试跟进滞后或实施欠充分的，应先改善测试然后才进入下一步迭代。

另外，评估机制还应尽量消除操作中出现的噪声，比如：测试评估范围可指定，非测试关注范围的源码不纳入测试程度评估，特定调用分支（如出错处理）经配置不纳入评估等。

4. 工具集成

持续集成过程涉及版本同步、版本比较，构建被测工程，实施回归测试、冒烟测试等操作，IT 工具应具备较好的定制扩展外部工具的能力，能将各种相关操作集成一起。

VcTester 共享版应用特色

VcTester 较完整的遵循第 4 代白盒测试方法的要求，在较高起点提供各项功能。下面我们从编辑器、调试器、测试功能、平台集成这四个方面分别评估 VcTester 共享版的功能价值，然后下一节在此基础上，补充说明商用版在功能增强后的应用价值。

1. 编辑器

VcTester 采用 CSE 的集成开发环境，它编辑器是采用 SynEdit 组件实现，与 Delphi 编辑器维持相同风格，也提供近似于 Delphi 的编辑功能，包括：提供与语法相关的关键字高亮显示，提示输入，定制的代码框架自动生成，支持较为完善的查找与替换功能，支持代码块缩进与缩出，此外该集成环境还支持函数与变量的定义跳转与回跳功能。

VcTester 的编辑功能强过 VC 不少，但比 SourceInsight 稍弱，稍弱的原因主要是 VcTester 在交叉索引、定义跳转方面没有 SourceInsight 做得彻底，另外，VcTester 编辑功能比较精简，SourceInsight 功能很全面，两者风格不一样。

2. 调试器

VcTester 共享版并未提供针对被测 C 代码的调试，只支持脚本的调试（具备基本的设断点、单步跟入跟出等功能），商用版才同时支持被测 C 代码与脚本的单步调试。

共享版只能借助 VC 实现单步调试，所幸的是，VcTester 的集成环境与 VC 并不冲突，只要将被测工程在 VC 中按 Debug 方式启动，然后使用 VcTester 发起调测，在 VC 中设断点进行单步调试。此方式主要缺陷是两者功能未在同一 IDE 环境集成，少不了要经常切换，对工具使用效率带来一定影响。

3. 测试功能

强大的测试功能支持是 VcTester 共享版最大的亮点。在线测试驱动（写测试脚本读写变量与调用函数）、在线测试桩在共享版已充分开放，如果再结合 VC profile，语句覆盖也有了，由此展开白盒测试已基本完整。共享版已较好的支持第 4 代白盒方法所要

求的在线测试功能，用例的调测效率大为提高，该模式下测试设计所见即所得，很直观，符合多数开发人员的编程习惯。

4. 平台集成

VcTester 共享版支持的工具扩展能力与商用版无差别，我们将 VC 的 Build 操作集成进来，使编码过程中用户可以随时按一个快捷键就编译被测工程，另外还可以集成 PcLint 代码检查功能，集成 Visual Leak Detector（VLD）内存泄露功能。VcTester 还针对捕获的 VC 编译或 Pclint 检查后的报错打印信息，提供快速跳转到出错源码行的功能。所以，VcTester 共享版虽然免费，但它提供的功能着实不弱。

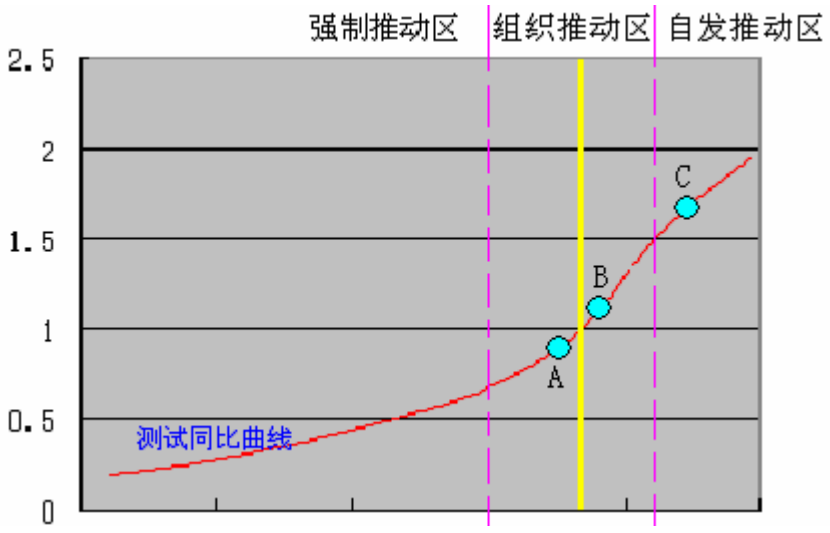
此外，还可以把 VSS 版本机操作集成到 VcTester 中，此举可有力支持每日构建、冒烟测试等持续集成实践。

外部工具集成也是 VcTester 的一个亮点，它提供可扩展能力并不逊色于 SourceInsight 平台。就上面举例的被集成功能中，Pclint 是商用工具（售价约 200 美元，当然网上也有不少破解版本），Profile 是 VC 专业版和企业版附带的，VLD 是免费且开源的工具，VcTester 共享版自然也是免费的，所以我们能以非常低的成本构造出功能强劲的持续集成研发平台。

总之，VcTester 共享版已能较好的支持嵌入式软件的持续集成实践，突破测试效率门槛，其应用效果已与业界主流的商用白盒工具相当。值得一提的是：VcTester 似乎只在“写一点测一点”的模式下才用得更好，比如提示输入、定义跳转等功能只在被测程序运行起来才具备，如果不是写一点测一点，被测程序都无法运行，提示输入、跳转等功能就没法用，导致整体易用性下降。再有，据已有实践反映，在 VcTester 下实施测试设计先行（Test Design First）相对容易，因为在线测试的所见即所得特点，让未见代码情况下写用例没那么让人感到惶恐。

更深入的功能

VcTester 共享版的功能已经很好了，想必商用版的功能更为强大吧？一点也没错，开发商对 VcTester 两个版本的定位如下图所示：



VcTester 共享版大致处于测试同比曲线的 B 点位置，商用版大致处于曲线的 C 点位置，A

点是目前多数支持嵌入式白盒测试的商用工具所处位置。也就是说，VcTester 共享版功能稍逊于业界知名品牌的商用工具，但在促进测试效率提升方面上仍优于大部分工具，与这些工具一样都处于“组织推动区”，当组织形态成熟、流程推动得力情况下是能将白盒测试做成功的，而 VcTester 商用版定位于再次大幅度提升测试效率，使白盒测试进入“自发推动区”。

VcTester 共享版适合于个体测试应用，主要针对小型开发，商用版则适合企业级应用，为适应大型开发下团队运作及产品质量保证活动而增加相应功能，商用版具有如下特色：

1. 支持符合第 4 代白盒测试方法的测试评估体系，包括：

   商用版提供 LICC 与 LDCC 两种代码覆盖率统计，对测试设计程度也提供评估，评估结果可以在线、直观的方式显示，评估标准可定制，另外还支持测试报告自动生成。共享版本没有这些功能。

2. 调测一体，支持将调试操作自动转化成测试脚本

   商用版的检视器支持调试操作转脚本，该功能可以促进大家养成自发测试的习惯，摆脱不自觉的被动测试状态，检视器还支持更强大的脚本桩功能，如条件桩、PreCheck 与 PostCheck 定义等。共享版没有这些功能。

3. 提供集成化的工作平台，可大幅提高开发效率

   商用版的源码与测试用例在同一个 IDE 平台编辑、维护，以相同形式同时支持测试脚本与源码的一体化调试，集成界面支持设置断点，进行单步跟踪。共享版本没有单步调试功能。

   VcTester 提供出色的 IDE 编辑器，编辑功能强大，支持提示输入、全文查找与替换、函数调用关系分析，定义与引用跳转、在线查看各行调用覆盖情况。共享版本没有函数调用关系分析与在线查看调用覆盖的功能。

4. 支持完善的测试消息构造与解析

   商用版提供用户数据 UDT 编辑器，可快速构造测试数据。共享版无此功能。

   商用版还将提供通用消息编辑器、消息解析器，可以自定义消息模板。该功能特别适合通信协议测试，其消息解析器与编辑器还可以免费集成到用户产品或相关 IT 工具上，借助本功能，用户可以将 VcTester 工具延伸到协议测试、功能测试等领域。共享版本不提供这些功能。

在提升测试效率方面，商用版的调测一体环境让整个持续集成的开发过程效率更高，另外，调试操作可转化为测试，强化了重用，使测试设计效率提升不少，还有在构造测试数据、自动生成结果检查语句等方面，商用版提供更多功能，都有力促进了测试效率的提升。

总结

最后，我们跳出 VcTester 提供的功能，讲讲这个工具的设计风格。总体而言，VcTester 的界面风格比较简捷、明快，没有繁杂操作过程，设置了不少热键，但并不是所有快捷按键都能在菜单或按纽得到提示，这对初学者多少有点障碍，但用熟后，多数用户都会喜欢上它的。比如，Alt+Up、Alt+Left、Alt+Right、Alt+Down 这 4 个快捷键设置，概括了最常用的几项

跳转操作，用熟了手感极佳，不像 VC 中此类快捷键的定义，键盘定位麻烦，用起来老觉别扭。

VcTester 的简洁风格还顺应了测试模式的要求，比如，文件目录树只一层，并不支持虚拟目录，之所以如此，是因为每位开发者或测试者，应该只工作在产品的一个侧面，VcTester 是假定用户只打开他所关心的某几个文件去编码与测试的。当然，这让工具缺失某些功能，容易招人垢病，但简洁风格维持下来了。

参考文献：

1. ezTester，Wayne Chan,《第 4 代白盒测试方法介绍（理论篇）》
2. ezTester，Wayne Chan,《第 4 代白盒测试方法介绍（VcTester 实践篇）》
3. ezTester，《VcTester 用户手册》
4. ezTester，《使用 VcTester 实施持续集成的组织管理模式》
5. ezTester，《使用 VcTester 构造持续集成及每日构建平台》