

完全自动化测试

VectorCAST/QA™

版权声明：本文档版权归创提信息科技（上海）有限公司所有，并保留一切权利。

功能：

- 持续测试
- 协同测试
- 自动化系统测试
- 代码覆盖率分析
- 基于修改的测试
- 代码变更影响分析
- 并行测试
- 提供 Web 界面的质量仪表盘
- 质量趋势分析

优点：

- 缩短了测试周期
- 加快了上市速度
- 提高了测试用例的质量
- 实现质量可视化

提高嵌入式开发质量的可预测性

利用 VectorCAST/QA，开发及测试团队就可以用一致的、可重复的流程来管理测试活动，并报告重要的质量指标。

为什么选用 VectorCAST/QA

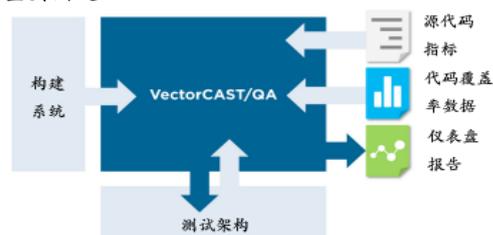
通常，软件开发测试小组面临的一个巨大的挑战就是：缩短新功能投放市场的准备时间。大部分软件开发及测试小组都有很多积压的新功能等待着发布。而产生这个问题的瓶颈往往都是：对待发布的版本进行完整的系统测试所花费时间太长了，一般都需要几天或几周的时间。

测试的时间太长，说明测试是在发布周期的晚期才进行的，而此前已经花了好几周的时间来完成复杂的集成工作。但是测试时常常还是会发现一些拖延发布时间的严重问题，以及让客户不满意的地方。通过 VectorCAST/QA，团队成员可以相互配合来完成进行测试活动，缩短测试时间，并能够提供最新质量指标，让大家及时了解发布的准备情况。

VectorCAST/QA 如何运作

VectorCAST/QA 可与用户所使用的构建系统和现有的测试架构相集成，来收集关键指标，如：代码复杂度，代码更新的频率，测试用例的状态，代码覆盖率数据。

使用 VectorCAST/QA **不需要**改变现有的工作流程或更换所使用的工具。在正常的系统测试工作开始的时候，VectorCAST/QA 会自动建成一个数据存储库，然后就可以回答一些类似于“针对这次代码更新，我需要做哪些测试？”这样的重要问题。



语句覆盖	已执行的测试	需要进行的测试	执行失败的测试
32%	6,798	14,175	2,921

VectorCAST/QA 仪表盘报告

协同测试

以前，软件测试都是由独立的测试小组来执行的，几乎不会共享测试用例和测试结果。所以，一般都是在发布周期的晚期才会发现很多 bug，因为这个时候才开始进行第一次测试。为了提高质量，必须要能够让研发团队的每个成员在任何时候都能够执行测试用例，而且测试结果要简单易懂。VectorCAST/QA 可支持协同测试，用户不需要学习新工具或新流程就能够执行测试。对于不同类型的测试，只需要配置一次连接器，以后就可以直接用来进行持续测试了。

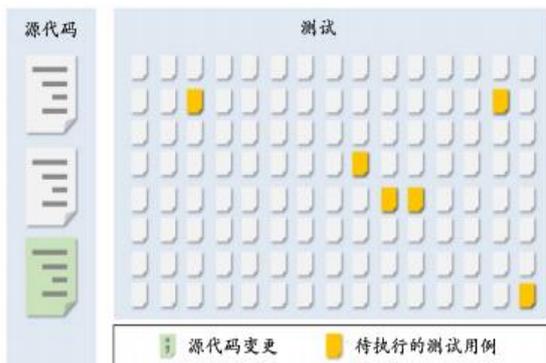
集成的代码覆盖率

要提高软件质量，第一步就是要了解当前测试活动的完整性。在测试过程中，VectorCAST/QA 可以自动获取并维护代码覆盖率数据，这就让用户能够迅速发现应用程序中未经检测的代码，从而确定还需要哪些资源来帮助提高测试的全面性。

基于更改的测试

缩短整体测试时间的关键在于提高测试效率。有些测试小组因为担心代码的修改会带来一些预料之外的影响，所以，他们在对经过修改的版本进行测试时，会将所有的测试用例都执行一遍，以保证所做的修改并不会导致质量倒退。但是，如果能够智能地选择最适合的测试用例子集来对所做的修改进行有针对性的测试会怎么样呢？这无疑会大大缩短测试周期，有助于更快地发布新功能。

VectorCAST/QA 的“基于更改的测试”这一功能正好可以提供这样的信息。根据从构建系统中收集到的数据，以及通过监控系统测试活动得到的数据，就可以智能地选出最小的测试用例子集，来所做的修改进行测试。VectorCAST/QA 能够发现测试和代码之间的关联。只要代码发生了改变，VectorCAST/QA 就能够自动计算出要对这次修改进行全面测试所需的最小测试子集。



代码变更影响分析

通过“代码变更影响分析”，可以确定源代码所做的修改对测试量产生的影响。这有助于开发人员在代码进行修改时，做出更好的决策。比如：开发人员可能会想将一个函数替换成另一个较低阶的常用函数。但是如果经过“代码变更影响分析”发现，该修改将至少增加 3 天的测试时间，那就可能会将该修改工作推迟到进度压力比较小的时候再进行。

测试用例的维护

测试用例和应用程序中的源代码一样，都是由开发人员编写的，每次添加了新功能之后，都会生成新的测试用例，所以随着时间的推移，就会产生很多难以理解的测试用例。如果测试过程中所用的测试用例不充分，就很难保证软件质量。但是，通常研发小组都没有很好地记录保存以前所用的测试用例，而且也没有随着应用程序的不断成熟，对这些测试用例进行改进。如果不了解被测试的内容或者什么需要被测试，那么现有的测试用例很可能就是浪费时间，并不能有效提高应用程序的质量。利用 VectorCAST/QA 可以看到每个测试用例调用的是应用程序的哪个部分，因而能够很好地评估每个测试用例的作用，并发现一些冗余的测试。

持续测试

虽然现在很多软件研发小组都利用类似于 CMake 的工具以及 Jenkins 这样的构建服务器解决了分布式和增量构建的问题，但是还是有很多研发小组需要进行持续和增量测试。VectorCAST/QA 有内置的用户自定义界面来连接到持续集成服务器，可以在多个虚拟或物理测试硬件上执行测试。将持续测试和“基于变更的测试”结合起来，能够大大缩短新 bug 的反馈时间，可由几天缩短到几分钟。

质量分析的关键切入点

新一代企业获得成功多倚赖实时、准确的数据，并且能够将这些数据转化成可有效运用的情报。利用 VectorCAST/QA，开发人员和质量检测工程师能够通过单一界面控制所有的测试活动并获取有意义的信息，这些数据对于制定决策来提高软件质量非常有帮助。使用 VectorCAST/QA 的目的在于可以轻松回答以下问题：

- 已经做了多少测试？
- 还要做哪些测试？
- 新版本可以发布了吗？