

为自动化测试 选择最佳的软件工具

陈宇睿
NI技术市场工程师

仪器的演变过程

电子管

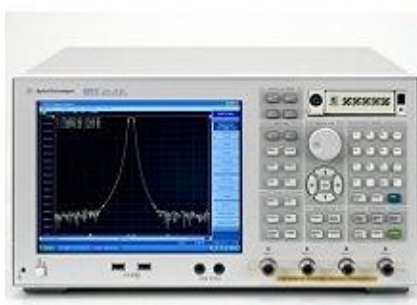


General
Radio



1920

晶体管
(集成电路)

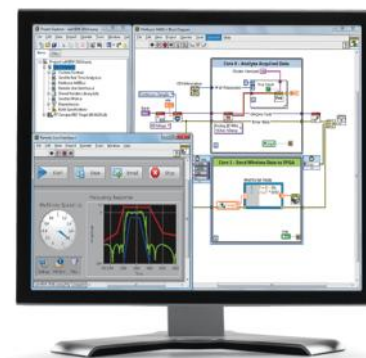


Hewlett
Packard



1965

软件

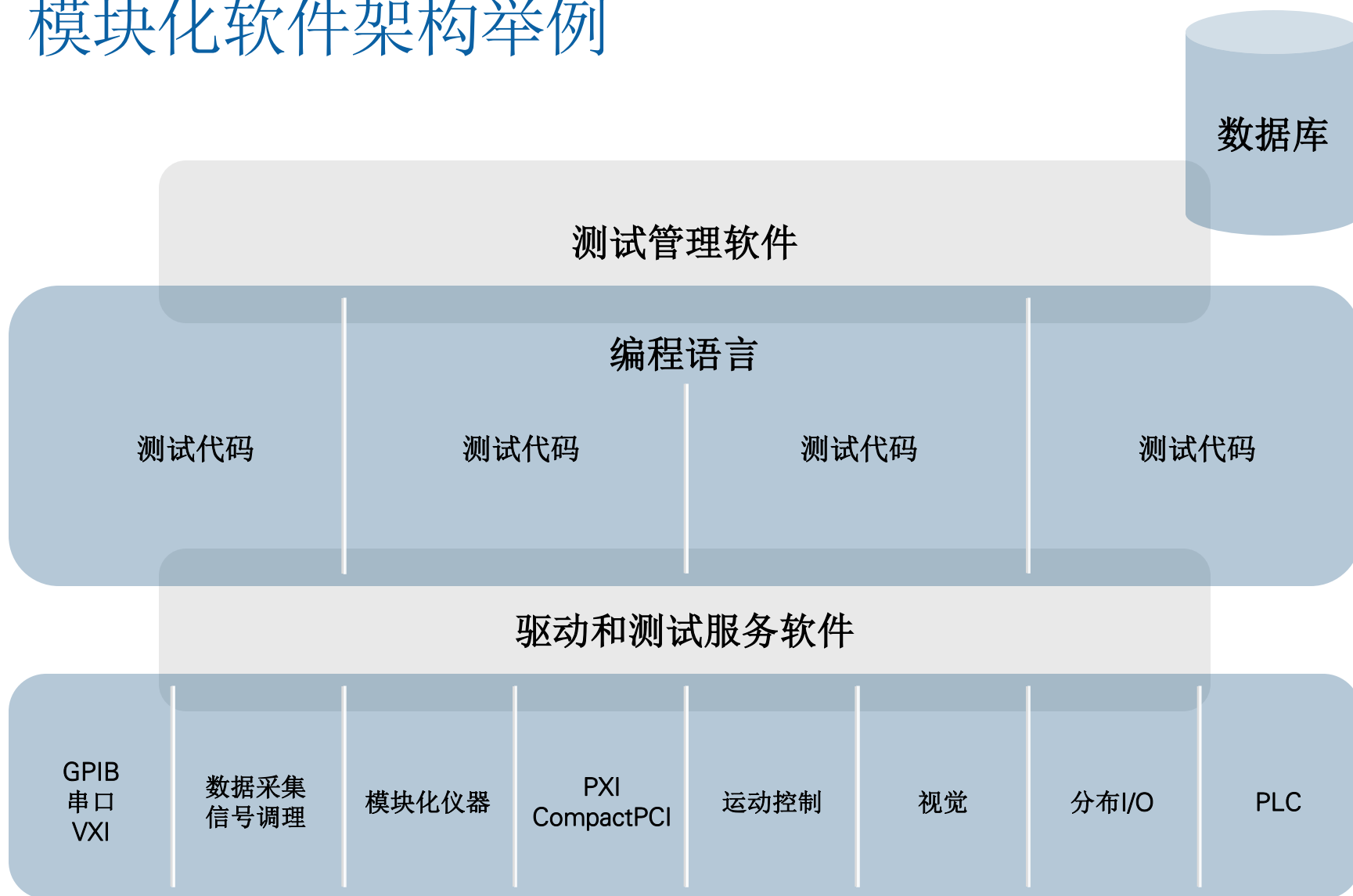


National
Instruments



2010

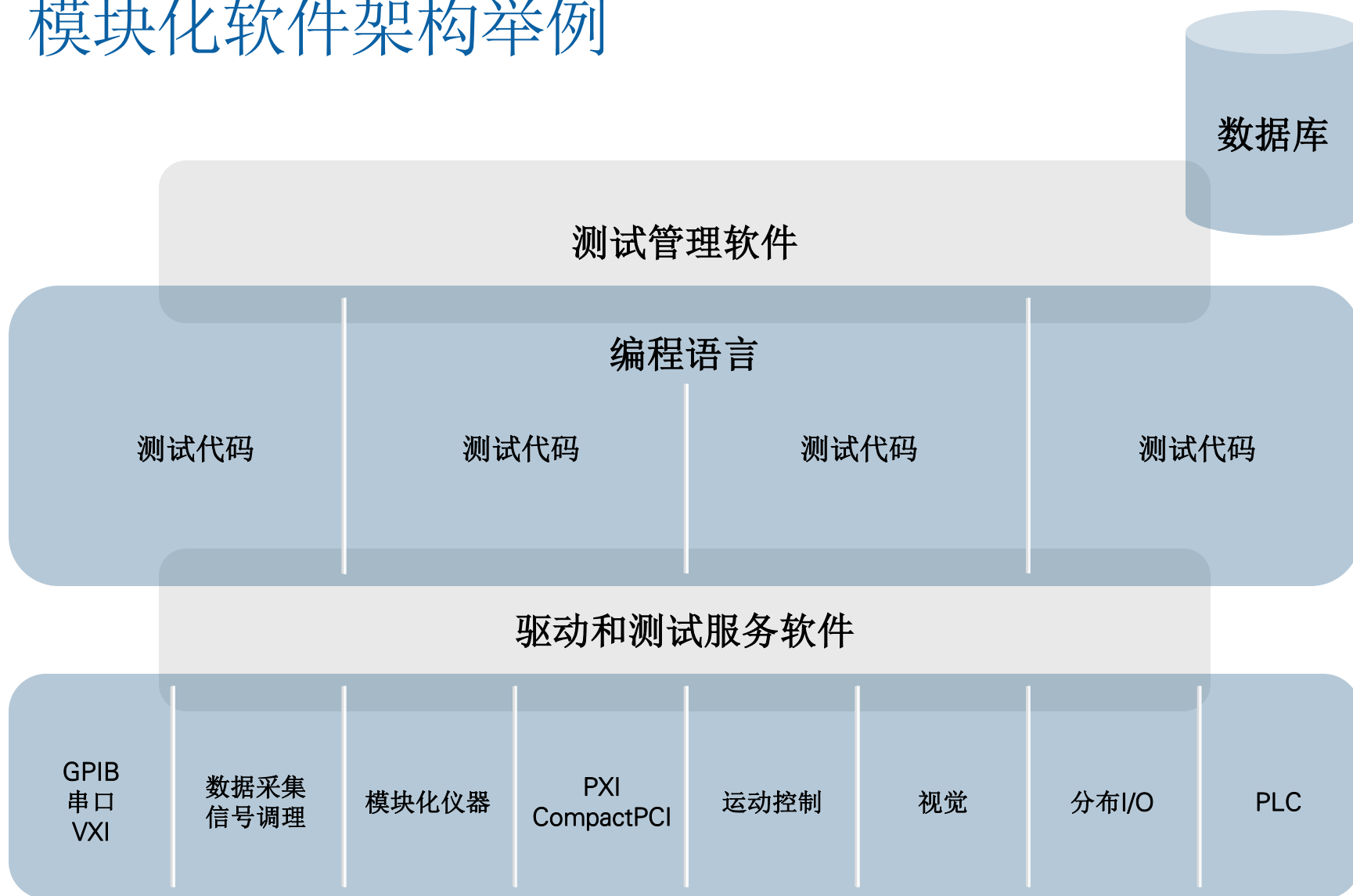
模块化软件架构举例



为什么要模块化

- 模块化组件的特点
 - 目的明确
 - 低耦合
- 模块化组件的好处
 - 更易于修改和维护
 - 便于重用
- 模块化测试系统的好处
 - 更灵活地适应新技术
 - 更容易地拓展新功能
 - 降低硬件被淘汰的风险

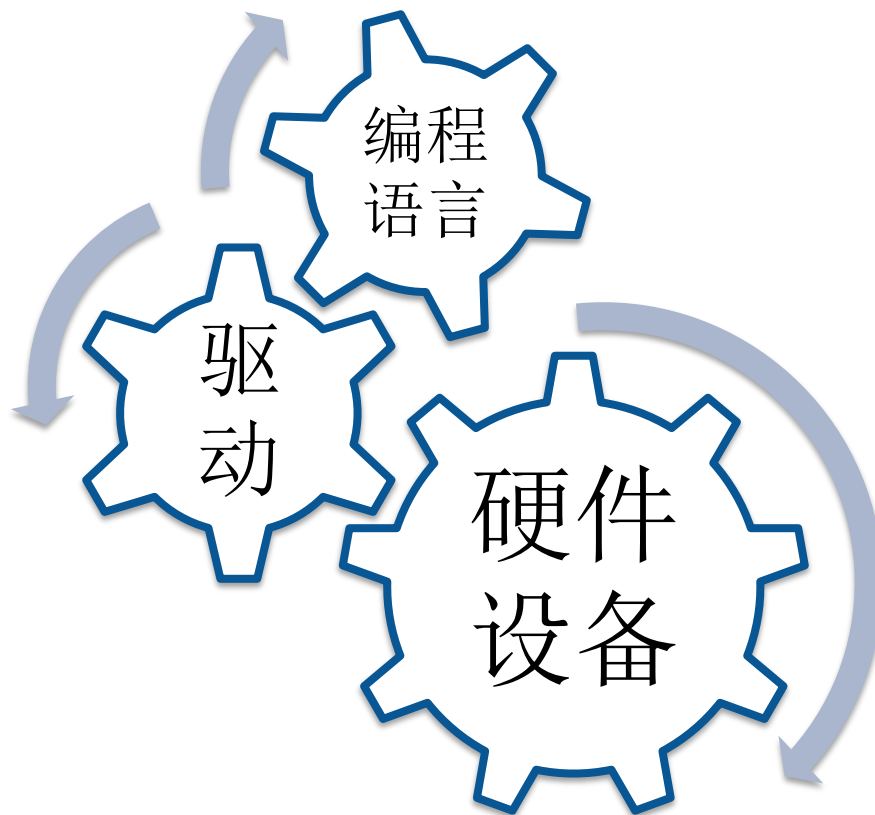
模块化软件架构举例



1. 驱动和测试服务软件

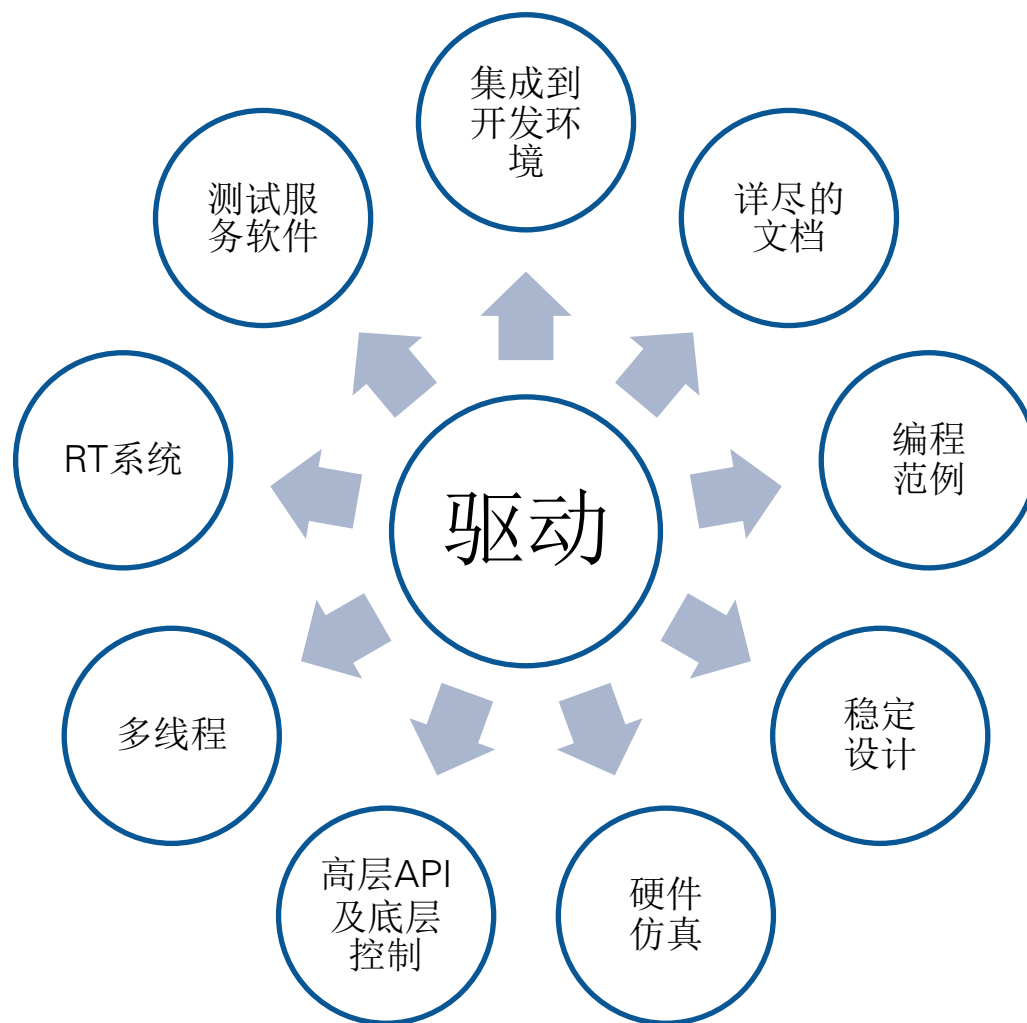
选择硬件平台时就要考虑驱动

- 驱动选择应考虑
 - 硬件设备
 - 编程语言



1. 驱动和测试服务软件

好的驱动不仅仅是带帮助的动态链接库(DLL)



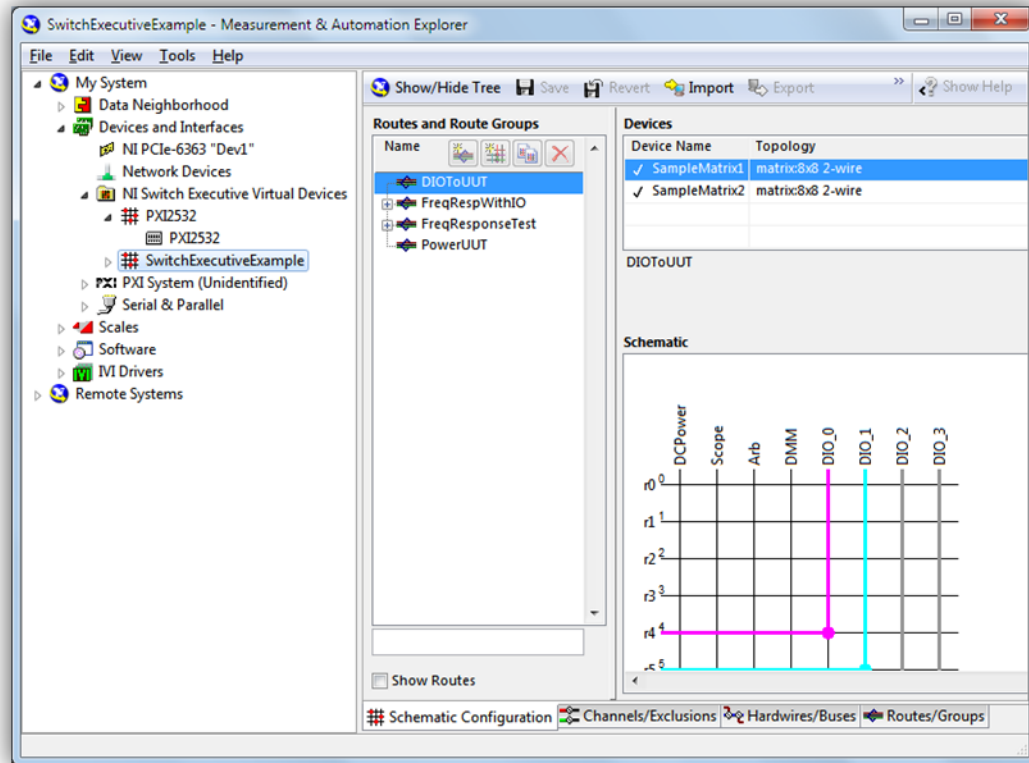
1. 驱动和测试服务软件

测试服务软件提供更多的工具

测试服务软件

-更多的硬件管理工具

- 罗列硬件
- 配置工具
- 自诊断工具
- 测试面板



Example: NI Measurement & Automation Explorer (MAX)

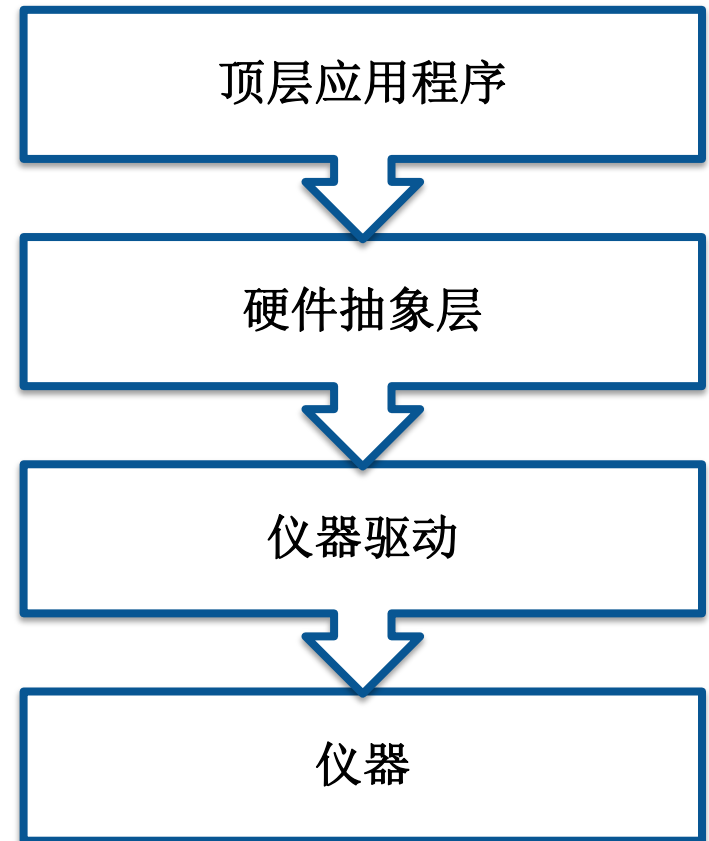
DEMO

1. 驱动和测试服务软件

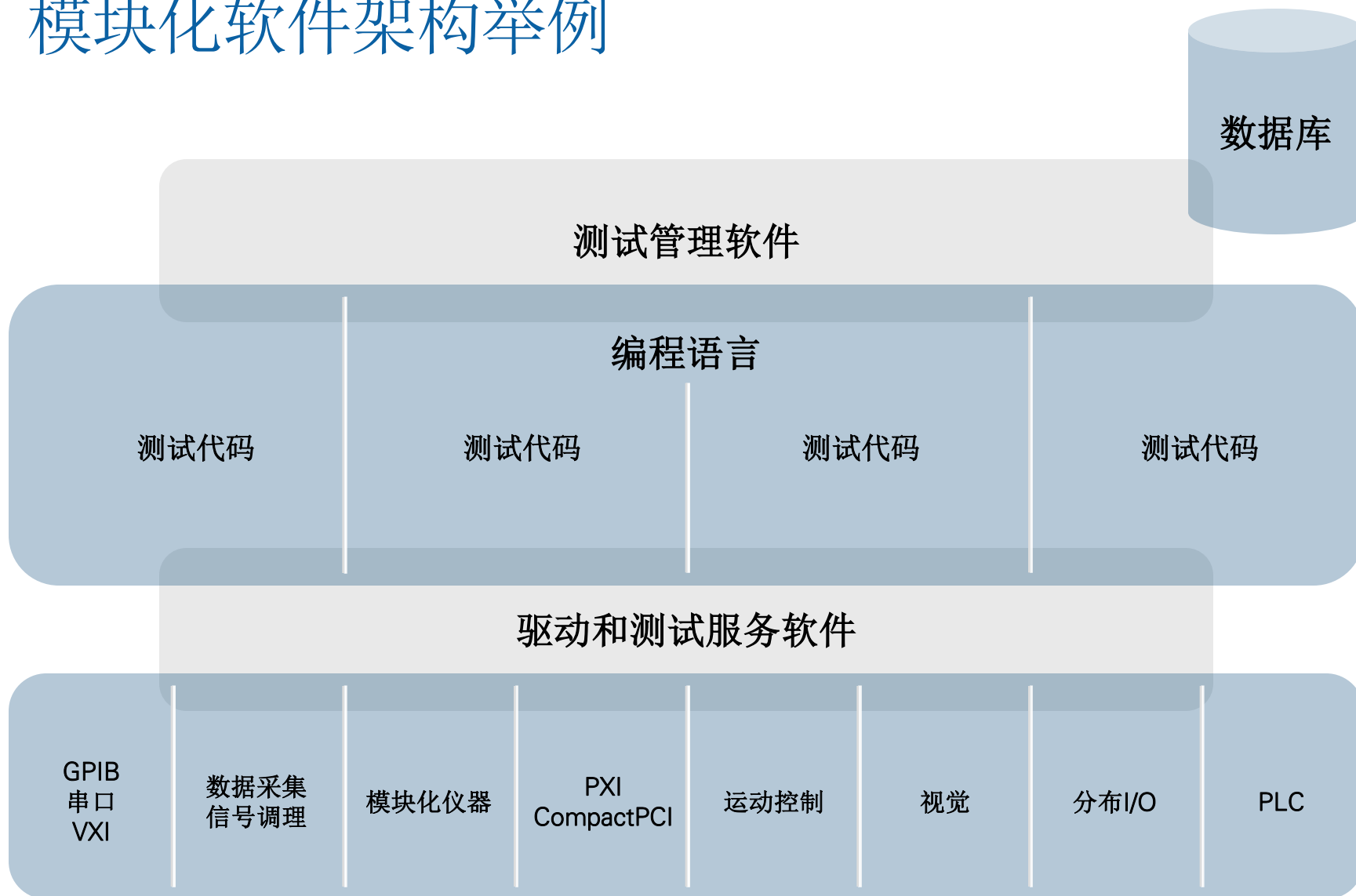
硬件抽象层在底层设备驱动之上提供有利的抽象层

硬件抽象层

- 是一个更高层次的驱动层，不受特定设备限制
- 有助于减少硬件淘汰
- 由于使用 标准化接口，因此更容易编写代码
- 例如：IVI
（Interchangeable Virtual Instrumentation）、
Robotics Sensor Driver

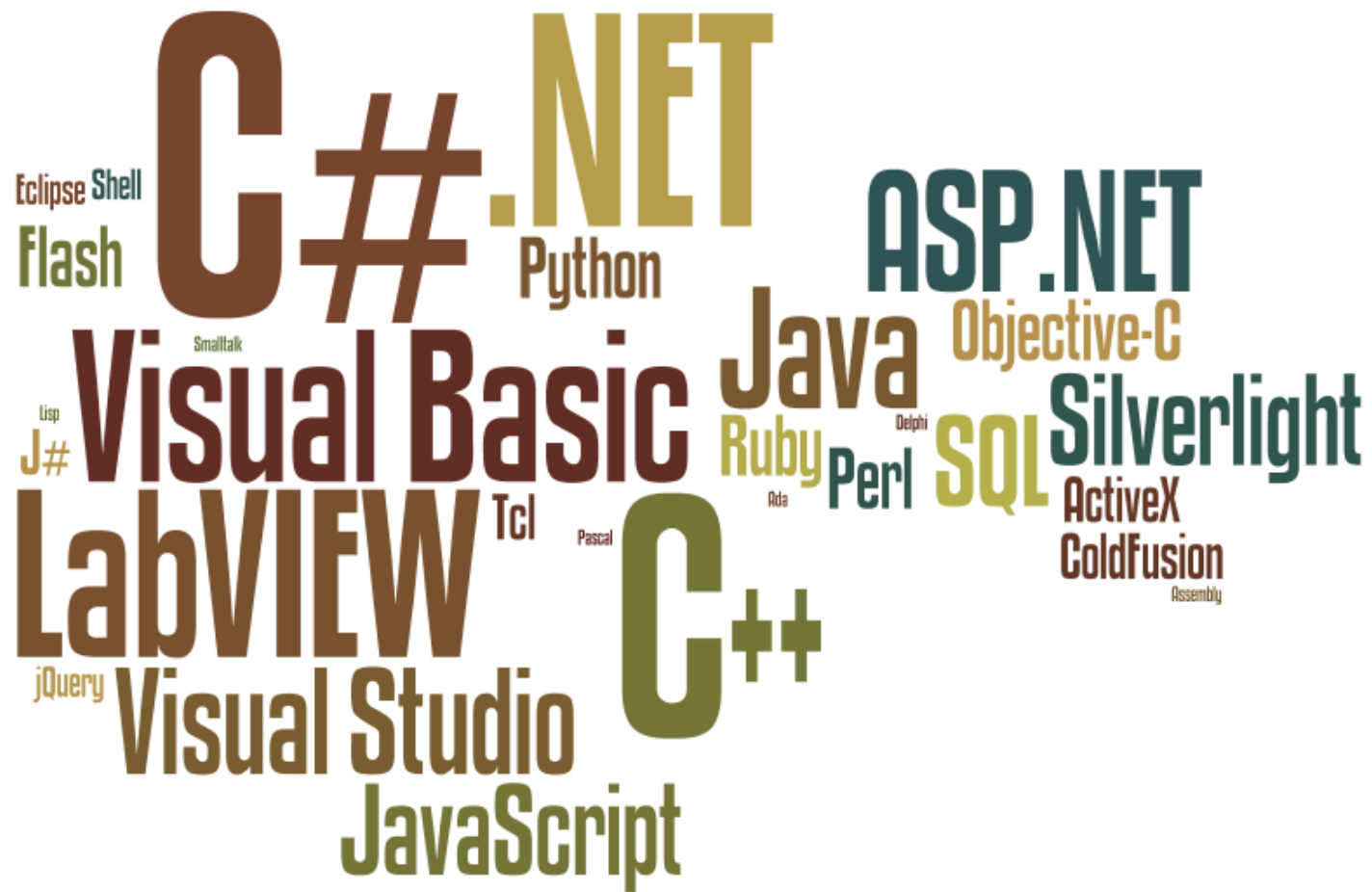


模块化软件架构举例



2. 测试代码（编程语言）

您有许多许多的选择



2. 测试代码（编程语言）

您有许多许多的选择

如何选择编程语言

- 掌握程度
 - 开发者了解这个编程语言吗？
 - 开发者能获取到培训吗？
 - 开发者能得到外部专家的支持吗？
- “工欲善其事必先利其器”
 - 对当前任务来讲，该编程语言是否高效？
- 驱动支持

2. 测试代码（编程语言）

C和C++是最通用的，但对大多数人来讲它们也是最复杂的

C/C++

- 优势
 - 大多数软件工程师有这类的经验
 - 标准化的版本更便于认证（例如，ANSI C）
- 有待考虑
 - 过于底层——自找麻烦
 - 对于非编程人员来讲学习起来有些困难
 - 用户界面的开发具有挑战性
 - 多线程和同步相对复杂
 - 不是为测试测量领域量身定做的

2. 测试代码（编程语言）

相对于C和C++来讲.NET更容易，但不是为测试测量行业量身定做的

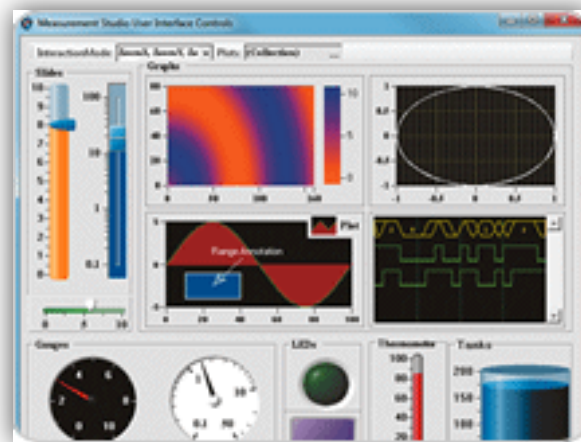
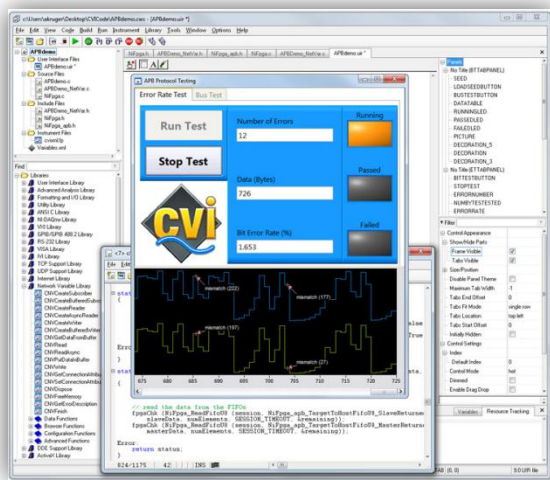
.NET（开发环境：Visual Studio）

- 优势
 - 大多数软件工程师有这类的经验
 - 和C/C++相比更容易学习
 - 用户界面更容易开发
 - 网络开发工具
- 有待考虑
 - 对于非编程人员来讲学习起来同样有些困难
 - 缺乏工程或科学方面的用户界面组件
 - 多线程和同步相对复杂
 - 不是为测试测量领域量身定做的

2. 测试代码（编程语言）

LabWindows/CVI, Measurement Studio 为C/C++/.NET的测试开发者提供一个优化的工具

- NI LabWindows™/CVI
- (ANSI C开发环境)
- NI Measurement Studio
- (Visual Studio中.NET的拓展组件)



为工程师和科研人员做的优化

- ✓ 工程/科学界面控件
- ✓ 分析函数库
- ✓ 集成驱动支持

The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

2. 测试代码（编程语言）

LabVIEW是一款针对于测试应用开发的、更易用的图像化编程语言

项目浏览器

管理和组织所有系统资源, 包括 I/O 和部署终端

部署终端

将 LabVIEW 代码部署至领先的桌面操作系统, 实时环境以及 FPGA 硬件设备

即时编译

在任意时刻及时观察应用程序的编译状态

前面板

创建事件驱动的用户界面控制系统和显示测量结果

计算模型

通过图形化编程组合和重用 .m 文件, C 代码和 HDL 代码

硬件连接

从任意仪器上的 I/O 将真实信号传递到 LabVIEW

并行编程

创建可并行运行的独立循环

程序框图

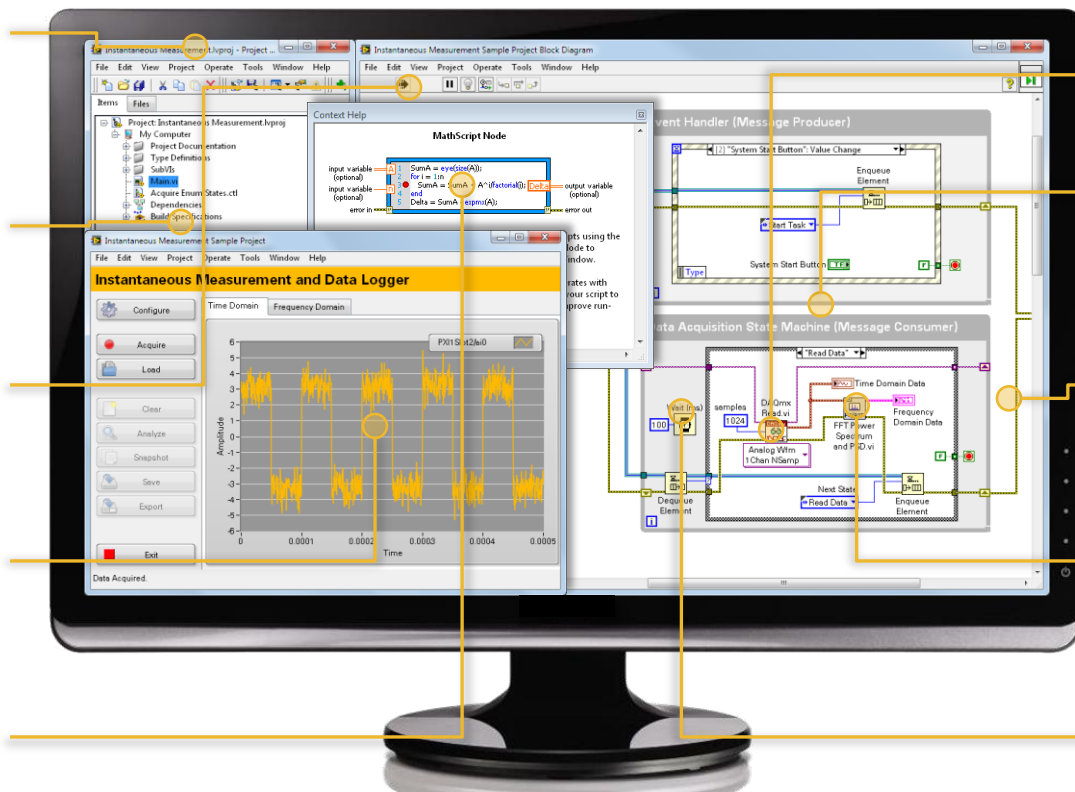
使用图形化编程自定义系统行为

分析库

专为工程师和科学家设计的高性能分析库函数

定时

使用顺序数据流定义外部执行顺序和定时



加速您的成功!

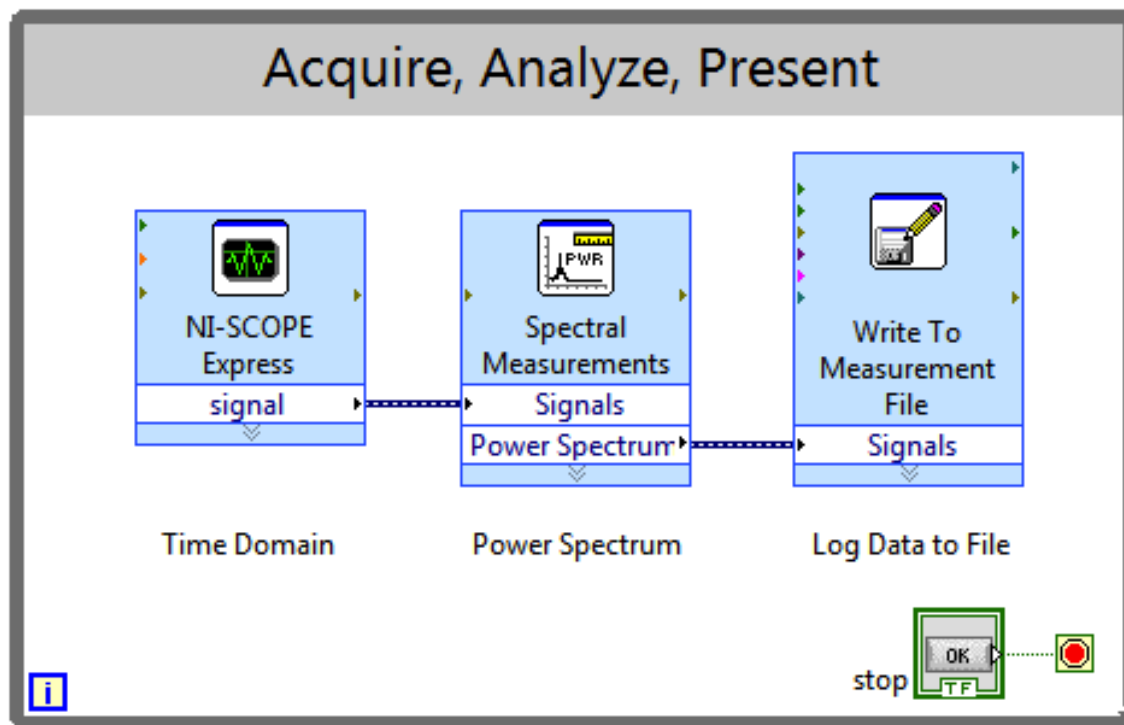
低级别的复杂性抽象和所有工具的整合, 助您构建任意的测量和控制系统

2. 测试代码（编程语言）

LabVIEW是一款针对于测试应用开发的、更易用的图像化编程语言

- 缩短测试系统开发时间

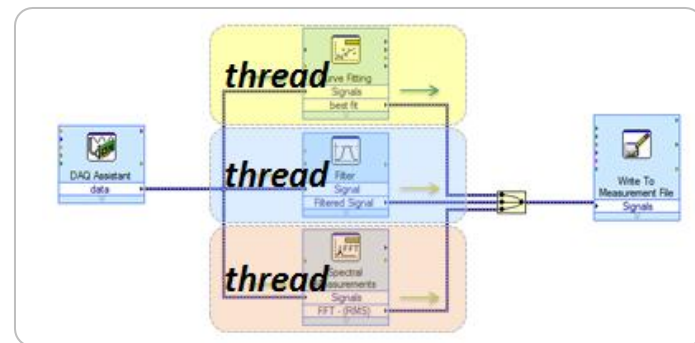
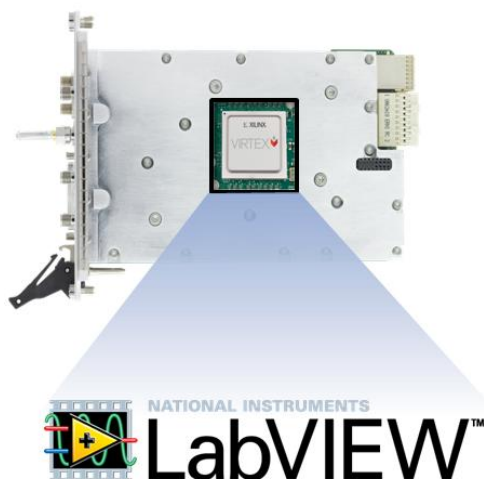
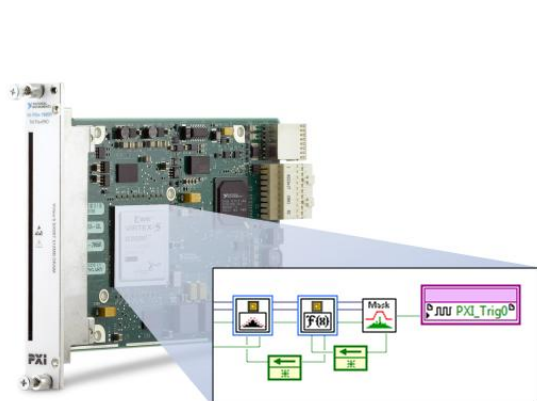
- 减少花在编程语法的时间，将精力集中在应对复杂的测试系统挑战，提高生产力



2. 测试代码（编程语言）

LabVIEW是一款针对于测试应用开发的、更易用的图像化编程语言

- 借助最前沿技术保持领先地位
 - LabVIEW利用多核处理器和FPGA等最新技术，提高测试系统的性能



2. 测试代码（编程语言）

LabVIEW是一款针对于测试应用开发的、更易用的图像化编程语言

- 轻松连接任何硬件
 - 在同一开发环境中实现所有测试系统挑战，提高生产力

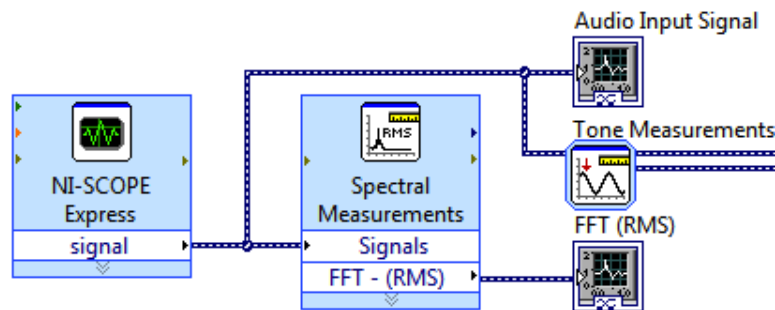
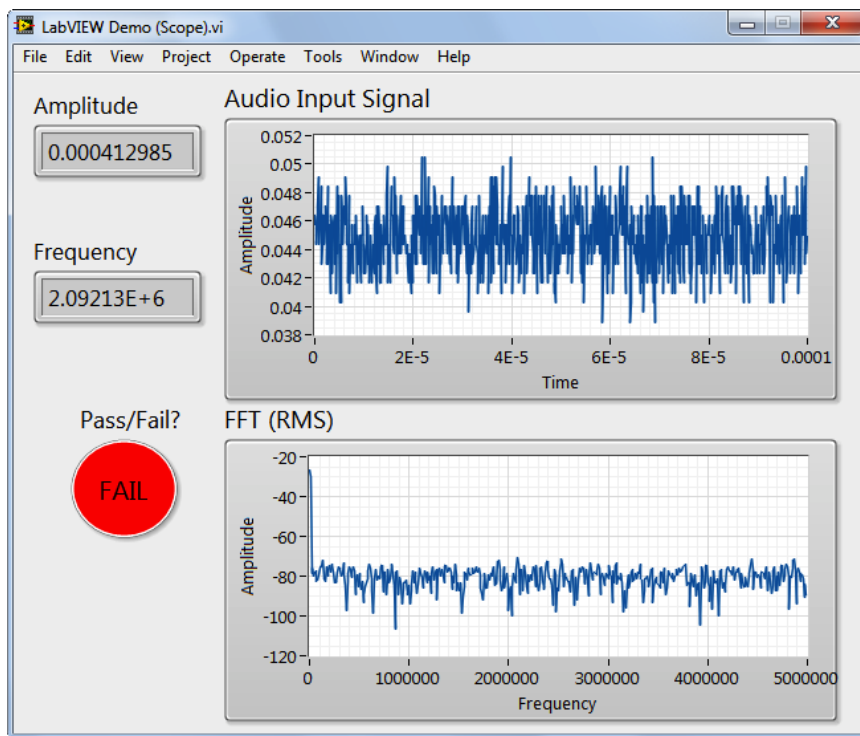


2. 测试代码（编程语言）

LabVIEW是一款针对于测试应用开发的、更易用的图像化编程语言

- 快速数据处理

- LabVIEW包含数千个专为测试工程师和科学家创建的高级函数

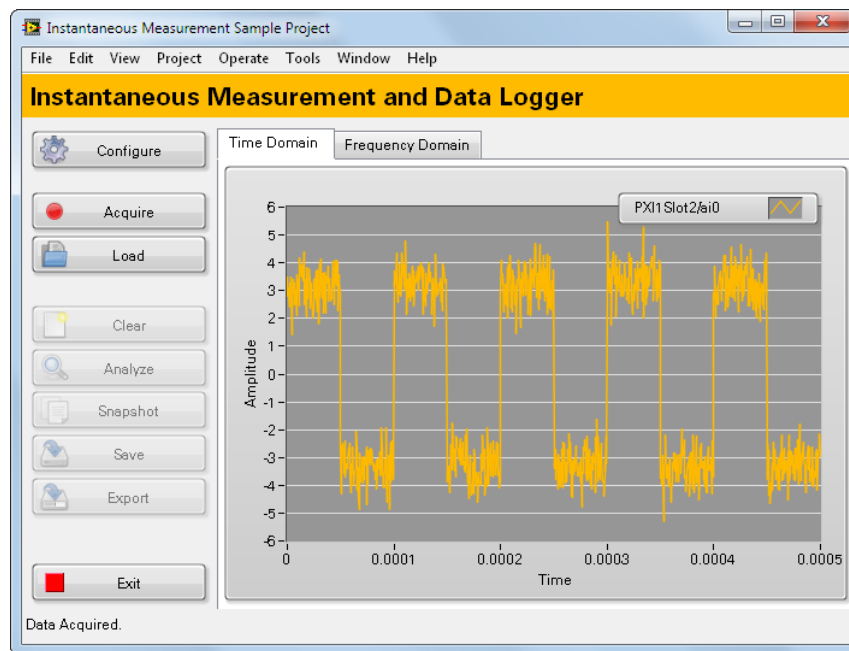
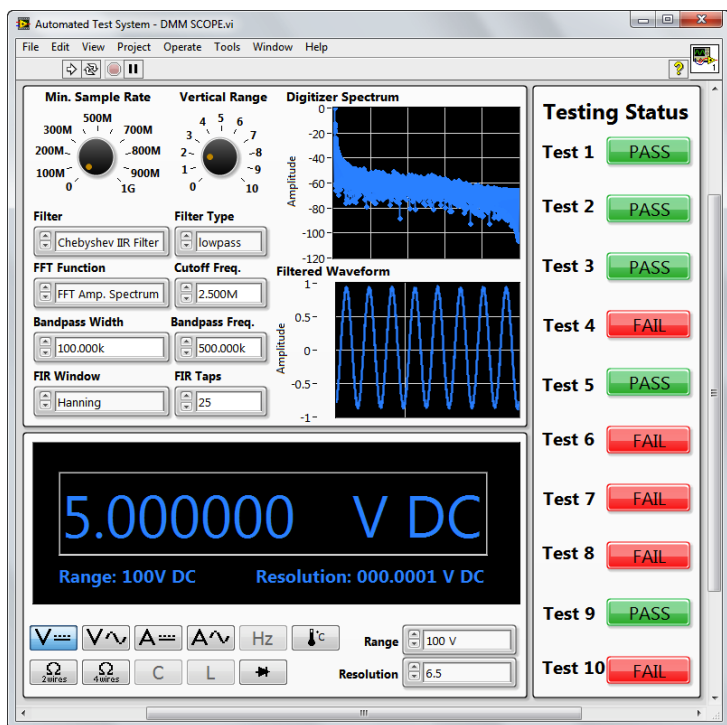


2. 测试代码（编程语言）

LabVIEW是一款针对于测试应用开发的、更易用的图像化编程语言

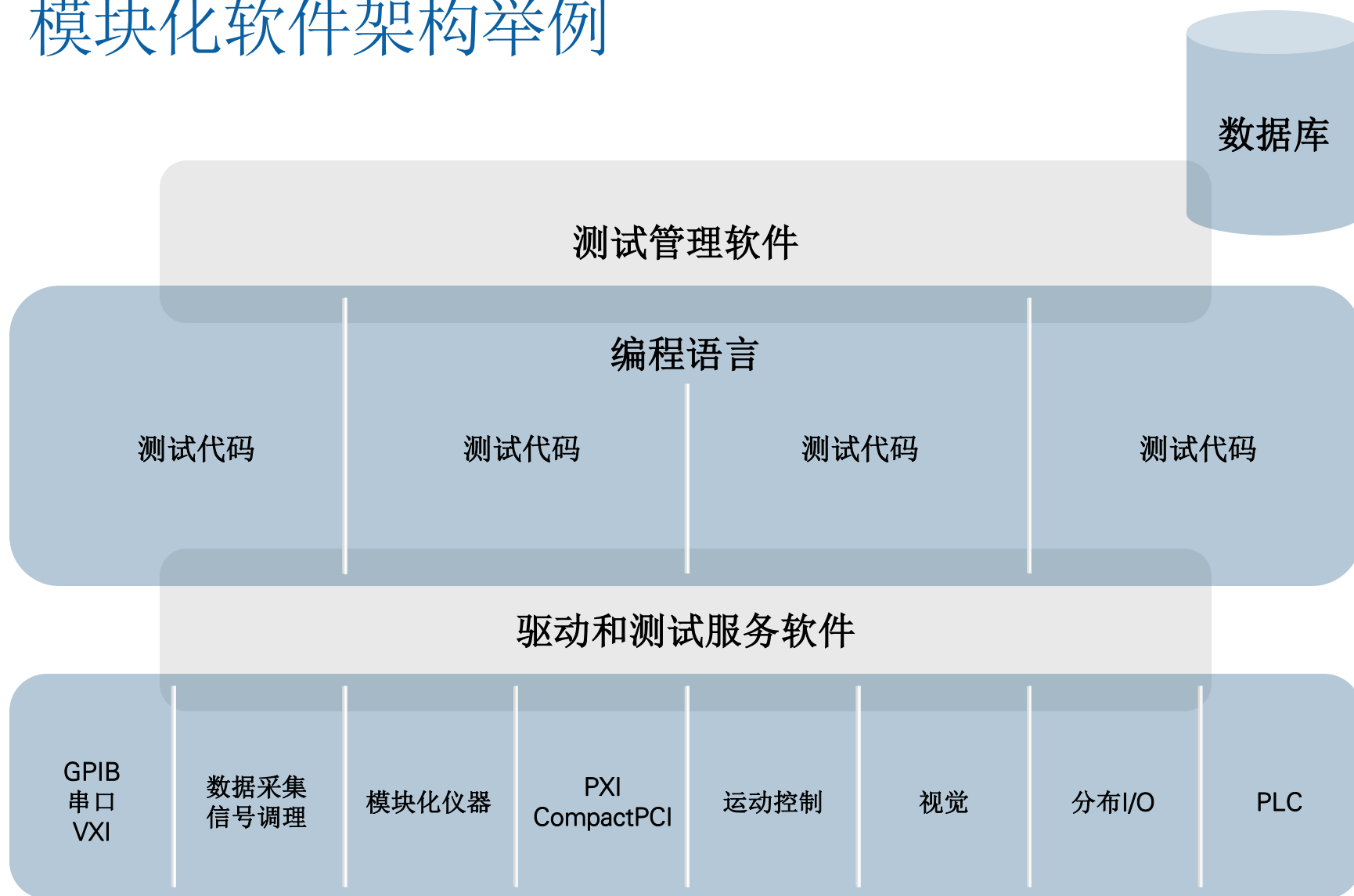
- 创建专业用户界面

- 通过拖拽的方式轻松创建用户界面



DEMO

模块化软件架构举例



3. 测试管理软件

测试管理软件的两个作用：排序和管理

什么情况下需要测试管理软件？

- 修改测试顺序
- 执行和调试测试
- 自定义用户界面
- 用户管理
- 处理结果（报告、记录）
- 测试流程控制

3. 测试管理软件

测试管理软件的角色

对每个被测元件
操作不同

- 校准
- 硬件配置
- 数据测量
- 结果分析
- 测试策略

对每个被测元件
操作重复

- 操作界面
- 用户管理
- 被测件跟踪
- 测试流程控制
- 测试报告

测试管理软件

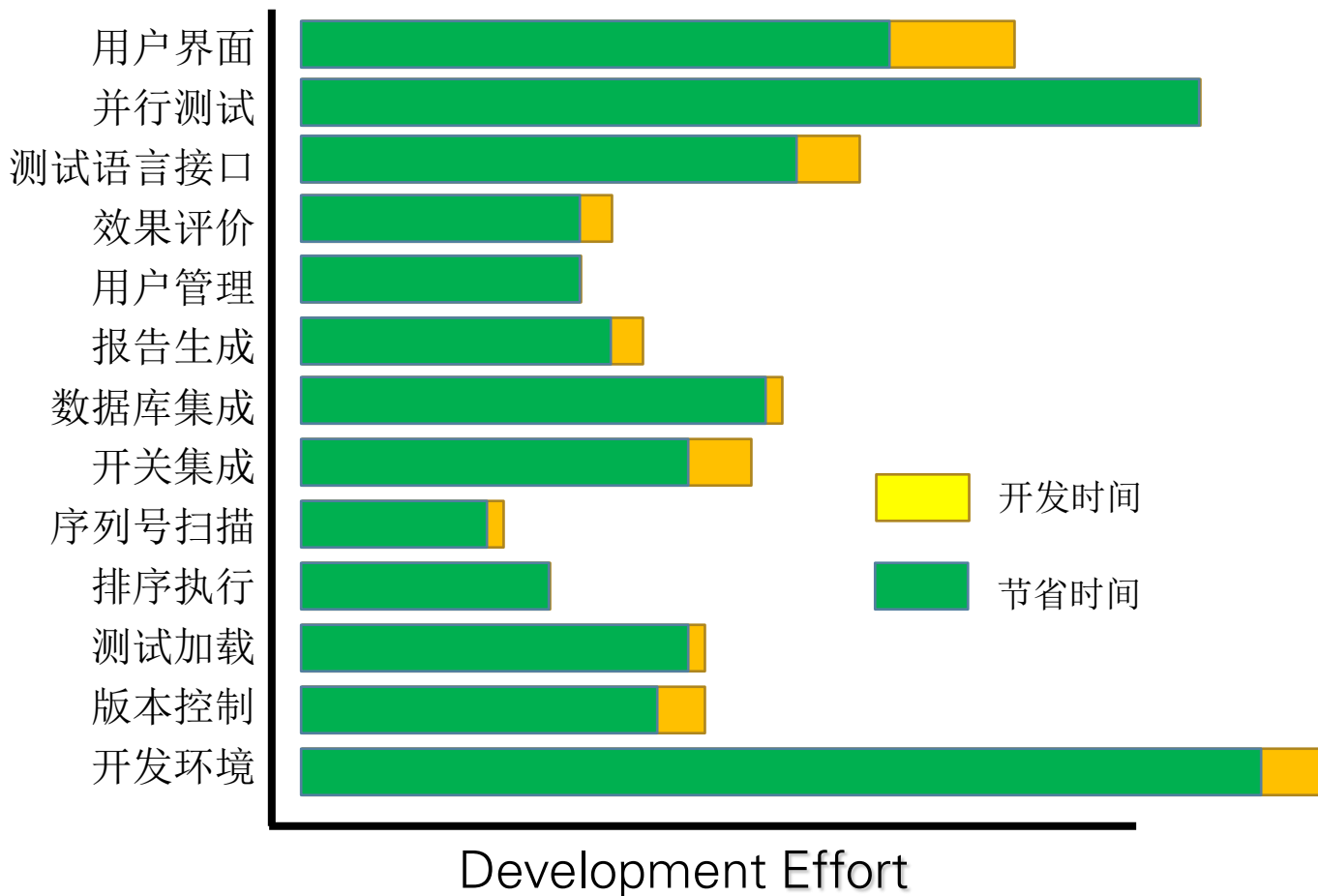
3. 测试管理软件

开发还是购买？

自己开发	直接购买
+ 没有单件成本	+ 随买即用
- 长时间的前期开发	+ 所有权成本非常低
- 所有权成本非常高	+ 行业验证
- 缺乏行业经验	+ 增加开发人员的可利用率
	- 没有源码
	- 增加前期软件成本

3. 测试管理软件

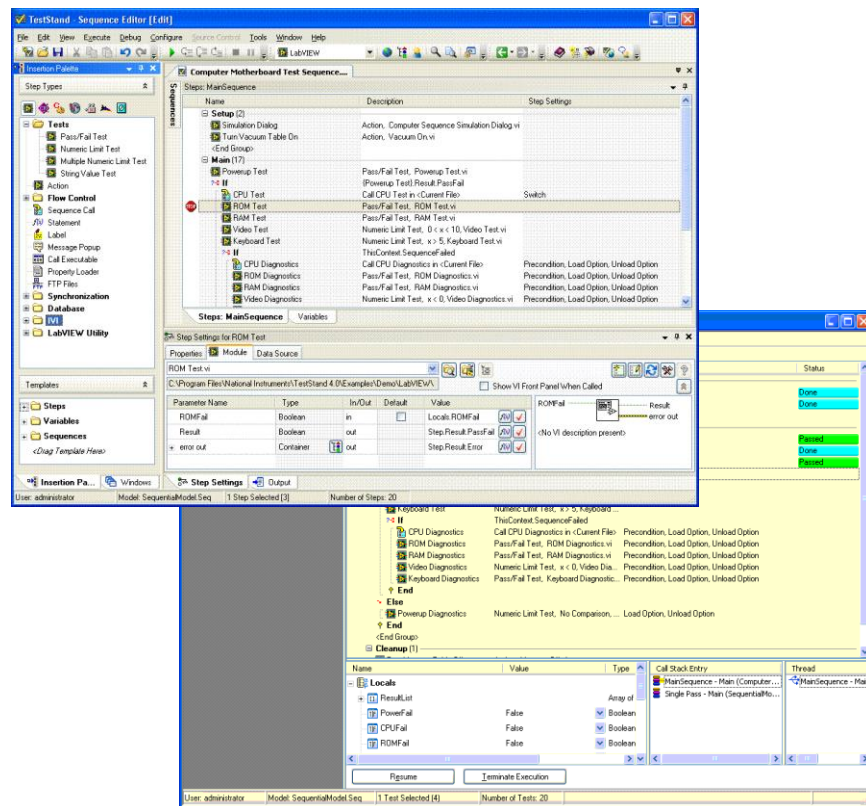
前期开发时间对比



3. 测试管理软件

NI TestStand: 随买即用, 可自定制

- 图形序列编辑环境
- 支持多种编程语言
- 多线程序列执行
- 多种报告格式
- 可直接与数据库连接
- 完全可定制
- 加速测试系统开发
- 简化测试系统的部署
- 提高测试吞吐量



DEMO

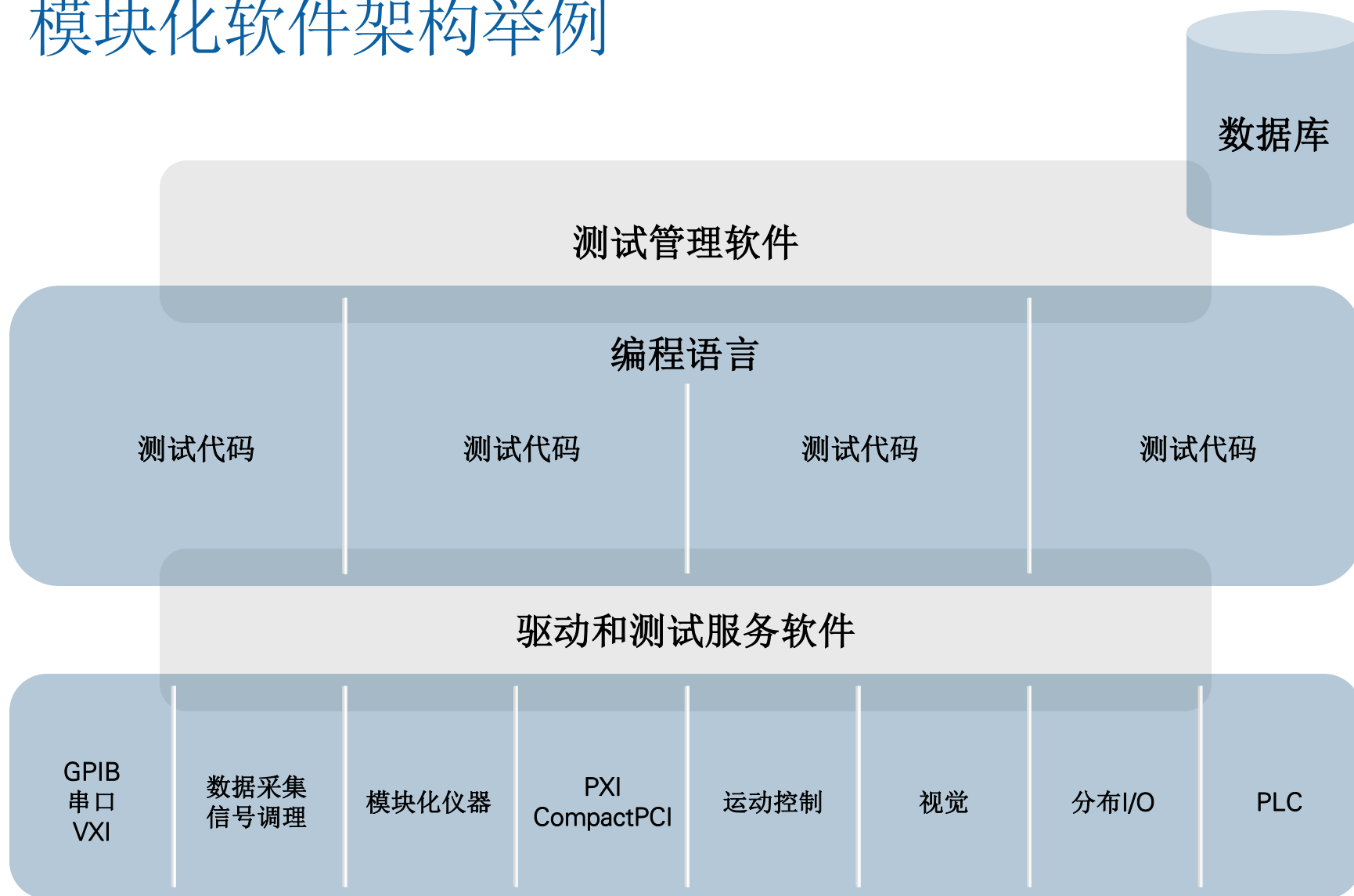
3. 测试管理软件

软件功能/回归测试不是为测试设备而制定

其他测试管理软件

- 软件功能/回归测试
- 实例
 - Microsoft Test Manager
 - IBM Rational Functional Tester
- 为软件测试而优化
 - 典型的单元测试,用户接口的测试,或回归测试
- 对设备验证/生产测试来说, 不是最优选择
 - 可以实现, 但这一过程转换会相对复杂
 - “工欲善其事必先利其器”

模块化软件架构举例



4. 数据存储与分析

从各种不同的行业标准中选择数据库

- 数据库是一种储存/存档测试结果的常用方法
- 最常见的是关系型数据库
 - 测试数据非常适合使用关系型数据库
 - 需要设计数据库架构
- 常用数据库
 - Microsoft SQL Server
 - Oracle
 - MySQL
- 成本、可拓展性、易用

4. 数据存储与分析

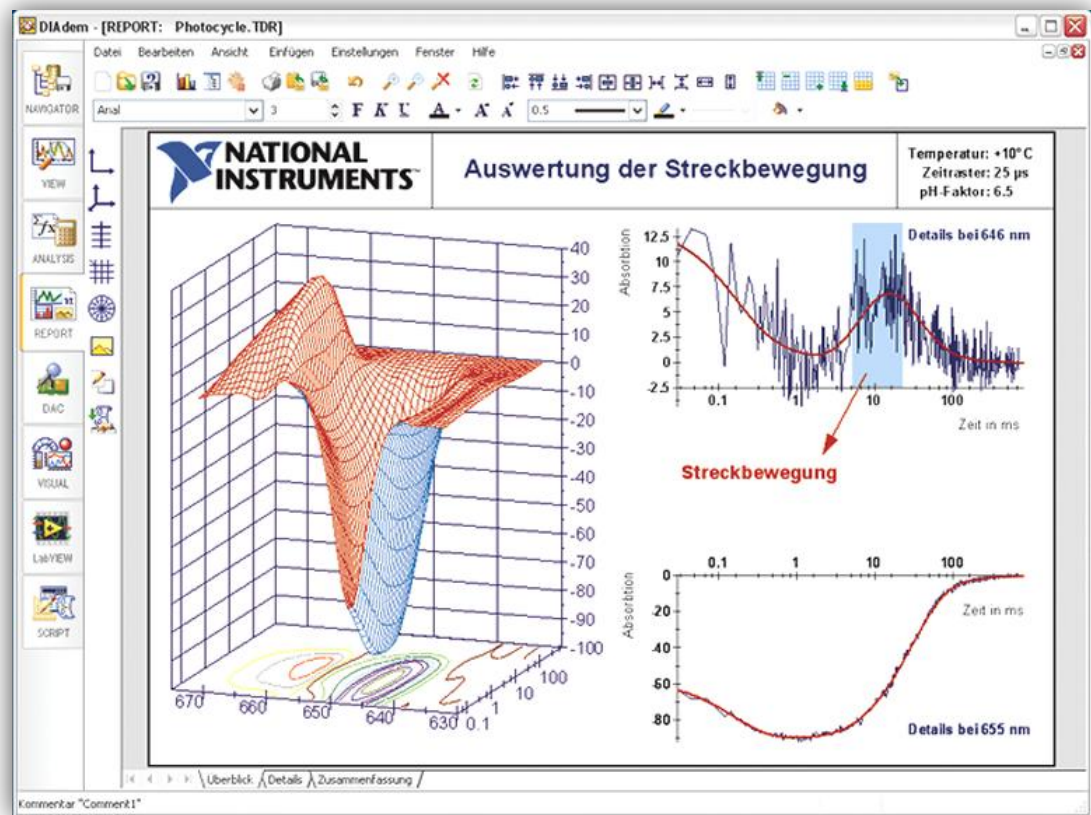
从各种不同的行业标准中选择数据库

- LabVIEW
 - TDMS
 - Database Connection Tool
- NI TestStand集成任何ODBC / OLE DB数据库
 - 包括上述所有常用数据库

4. 数据存储与分析

数据分析工具可以帮助您理解您的数据

- 使用数据分析工具来了解测试数据，并提高生产效率
 - NI DIADem



4. 数据存储与分析

数据分析工具可以帮助您理解您的数据

DIAdem

强大的数据搜索和挖掘能力

简单灵活得访问数据库和文件

交互式分析和报表生成

通过VB脚本实现自动化控制



4. 数据存储与分析

数据分析工具可以帮助您理解您的数据

DIAdem

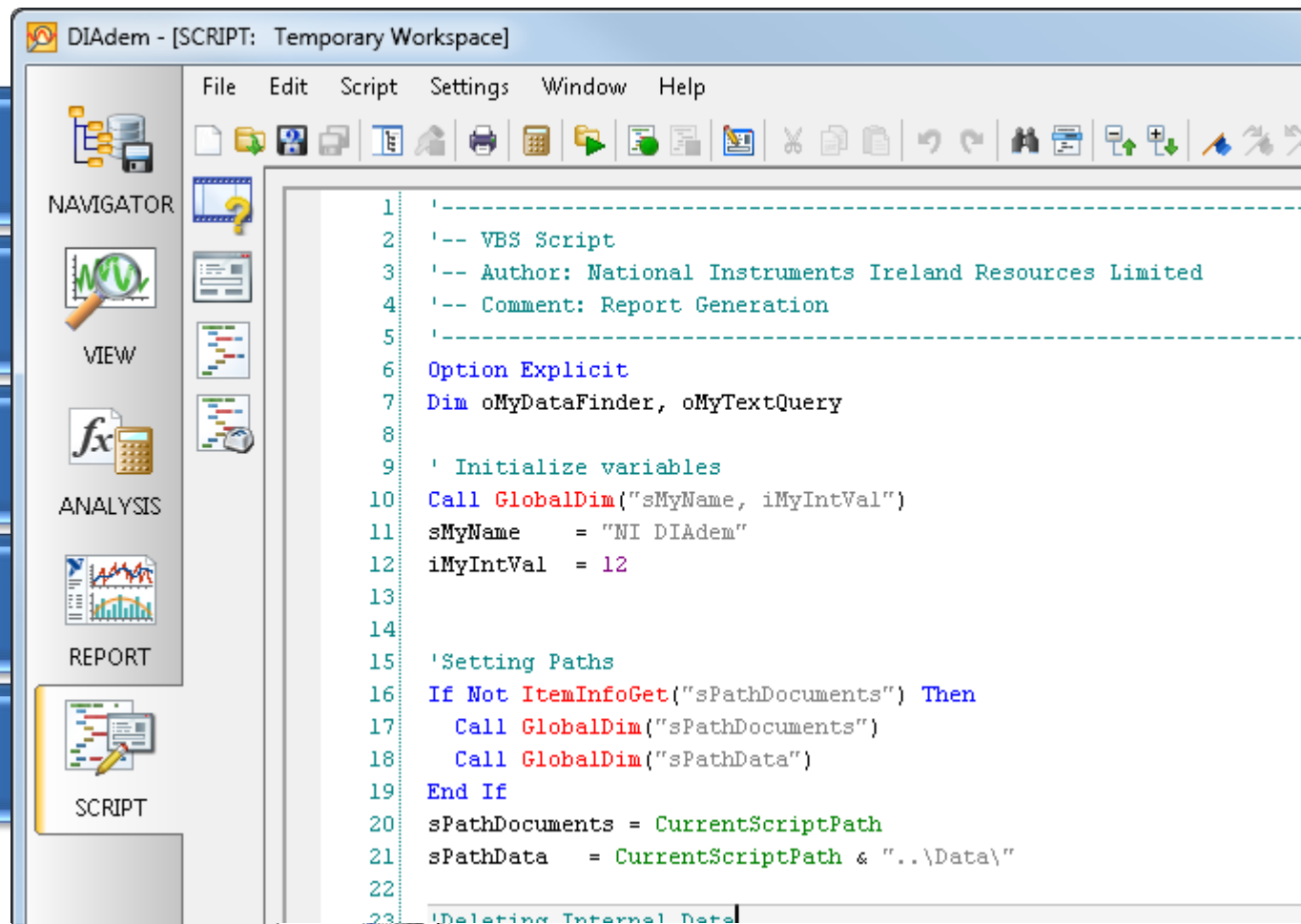
查询和加载数据

检查数据

分析数据

结果报告

自动化应用



5. 需求管理

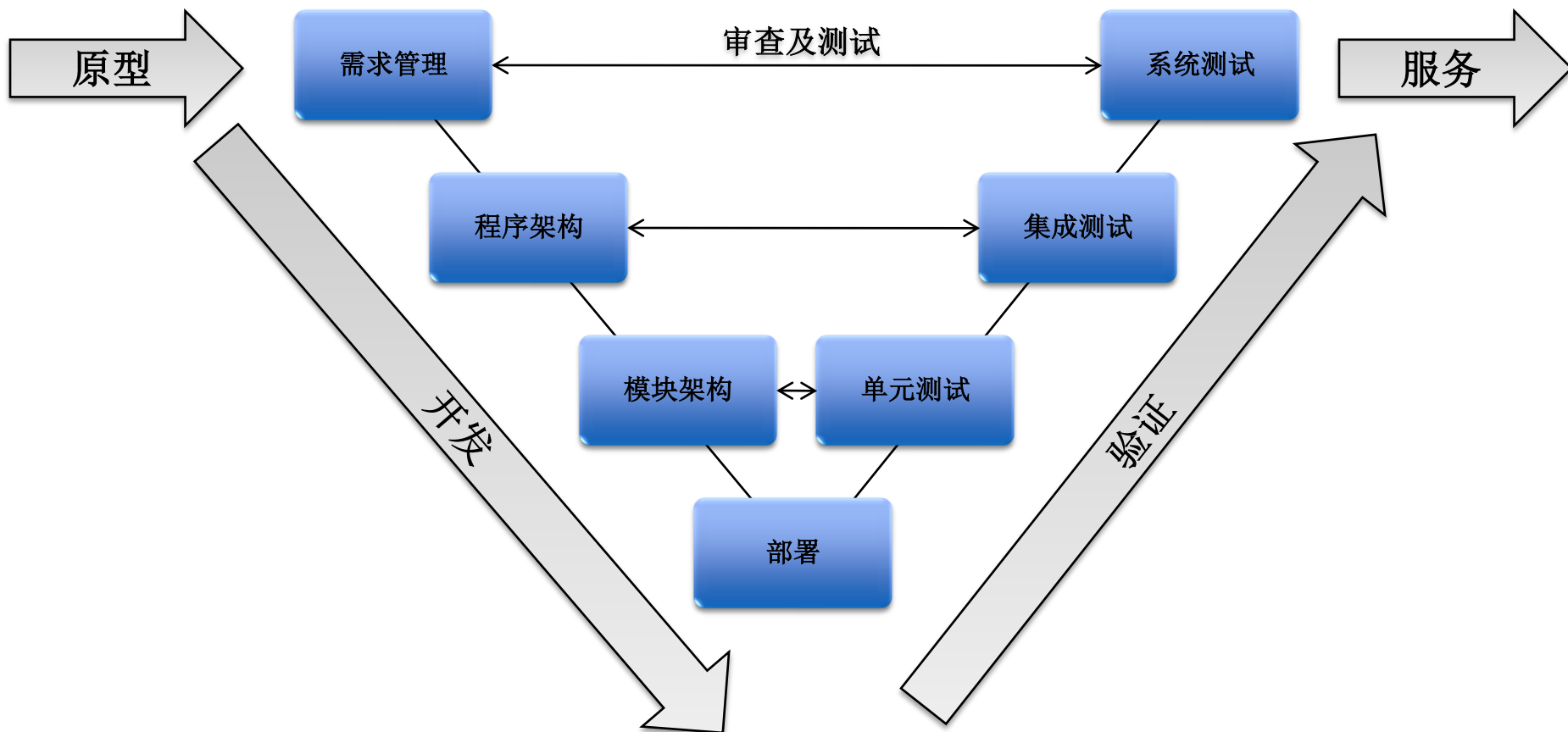
通过需求来验证应用程序的质量和完成进度

什么情况下需要需求管理？

- 项目复杂
- 行业标准要求
- 开发进程控制
- 开发工作繁冗
- 不确定因素

5. 需求管理

V模型

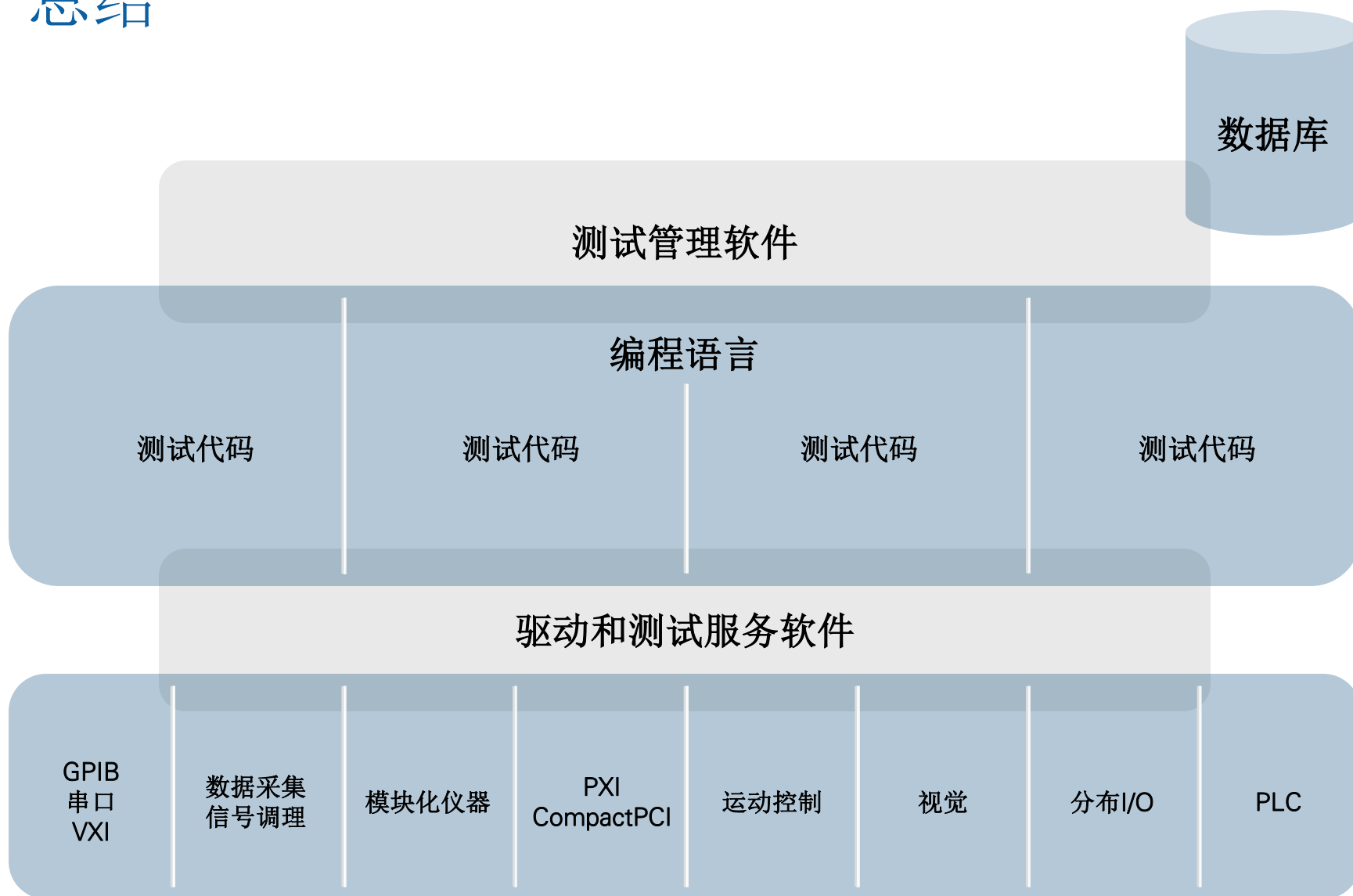


5. 需求管理

NI Requirements Gateway 追溯需求及其执行进度



总结



个案研究：泰雷兹通信公司

挑战

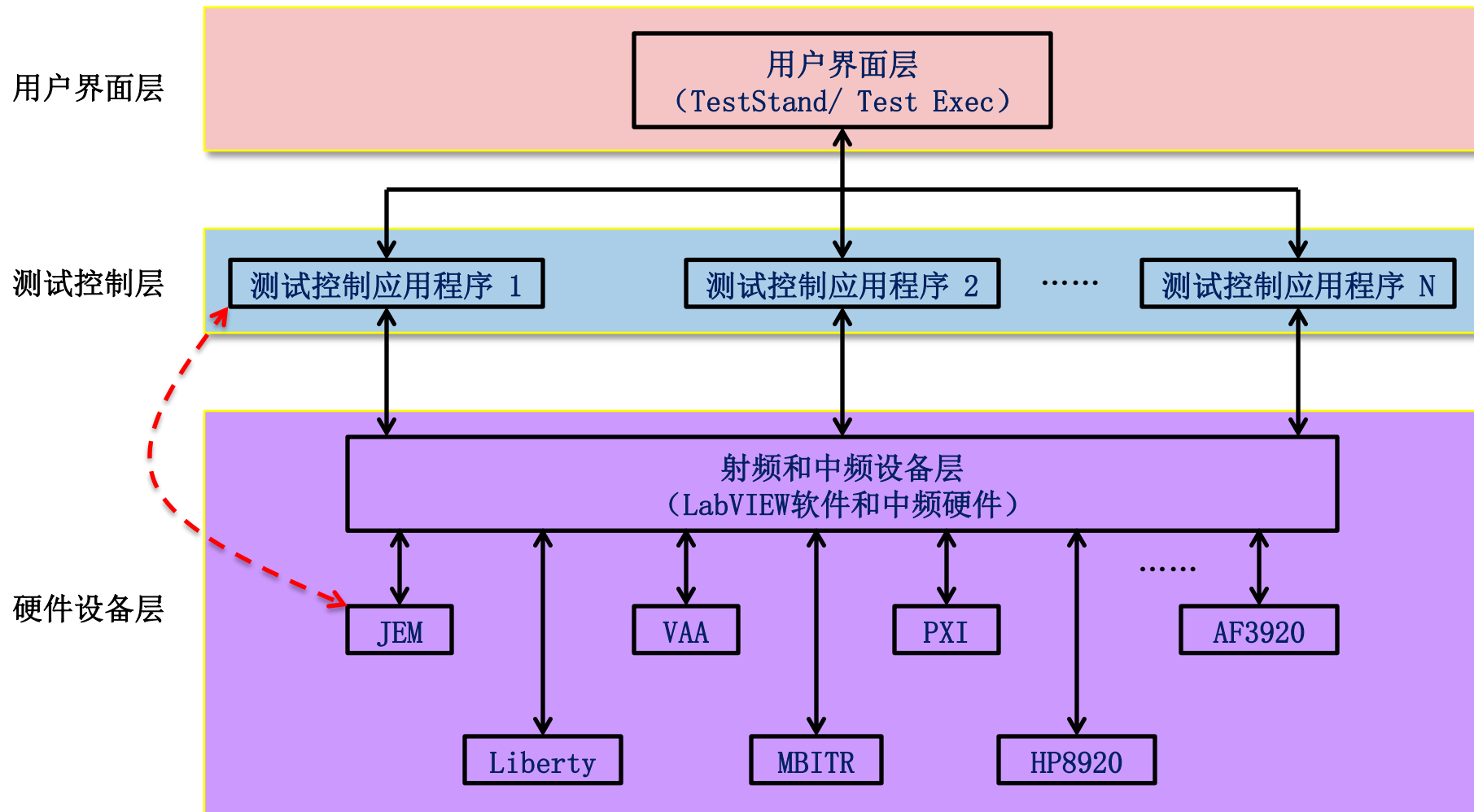
- 手动配置和执行的验证测试一直是缩短产品开发周期的瓶颈

解决方案

- 基于NI LabVIEW软件和PXI硬件实现一个自动验证测试框架，从而达到测试自动化和代码重用的目的



个案研究：泰雷兹通信公司



个案研究：泰雷兹通信公司

效益

- 验证测试时间减少92%
- 人工操作减少79%
- 超过100次的随机测试设计
- 通过提早发现不常见的差错提高产品质量
- 282%的投资回报
- 5.9个月的投资回收期
- 年节省数百万美元的测试开发成本

使用NI的测试平台，Flextronics大大节约成本

- 挑战:

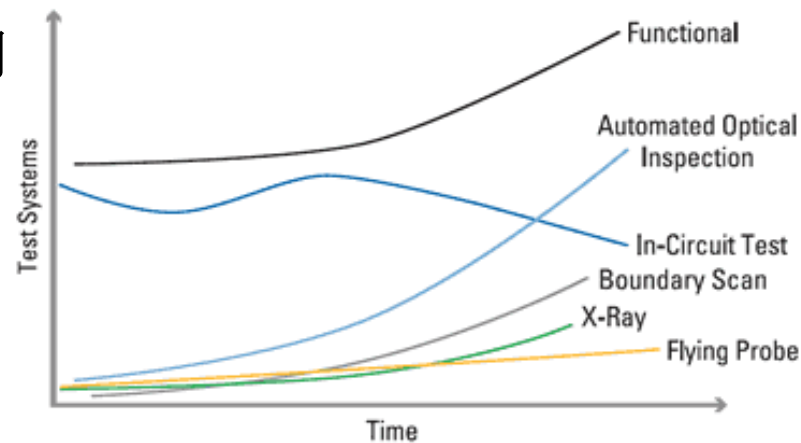
- 使用标准的、可升级的测试平台应用的日常测试

- 产品

- TestStand, PXI, 模块化仪器

- 应用

- 配合Flextronics的测试标准（FTS），无论任何测试人员，测试场所还是待测产品，NI通用的软硬件平台都能确保测试的正常运行。
- 使用NI的测试平台，Flextronics在日新月异的测试技术更新中大大节约了测试投入



来自Flextronics(中国)的用户感言

- “PXI平台真正革新了我们的自动化测试。通过PXI开放的架构，灵活的配置，低成本以及广泛的可供选择的模块化仪器，使得测试开发工程师更快更高效的开发测试应用。我们上一个项目因为使用了PXI，帮助节省了至少20%的设备成本，并且仅在3个星期内就完成了对PXI系统的开发。”
- - - Jay Mariano, Test Development Engineer,
Flextronics Test China

更多自动化测试与测量专题

- 11:15–12:00 会议厅3E
《为自动化测试选择最佳软件工具》
- 11:15–12:00 黄河厅
《纵览新一代数据记录系统》
- 13:30–14:15 黄河厅
《使用最新技术和总线进行高速、高吞吐量测试》
- 13:30–14:15 会议厅3J
《从概念到原型-基于LabVIEW的软件无线电平台介绍》
- 14:30–15:15 国际厅
《选择PXI平台的几点考虑》
- 15:30–16:15 国际厅
《多通道数据采集与分析系统的构建要诀》
- 15:30–16:15 会议厅3J
《基于LabVIEW FPGA构建高速流盘应用》

谢谢！