

**Automated portability testing with  
virtual test environments**  
*door Wim Demey*

---

**19 november 2009**

voordracht georganiseerd door het



**TECHNOLOGISCH INSTITUUT**  
*Discussiegroep Software Testing*

met de steun van



systematically delivering success



---

Ingenieurshuis - K VIV, Antwerpen

# Automated portability testing with virtual test environments

November 19, 2009

*Wim Demey*  
[wim.demey@ctg.com](mailto:wim.demey@ctg.com)

Copyright © CTG, Inc.



## Agenda

- Context
- Portability testing
- Solution
- Demo
- Conclusions
- Questions

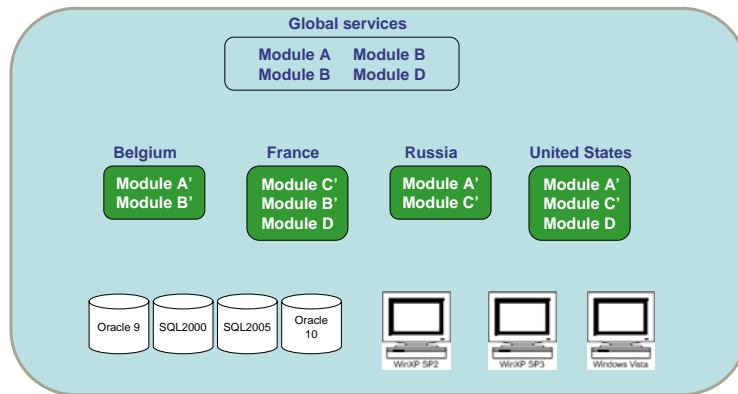
Copyright © CTG, Inc.

2



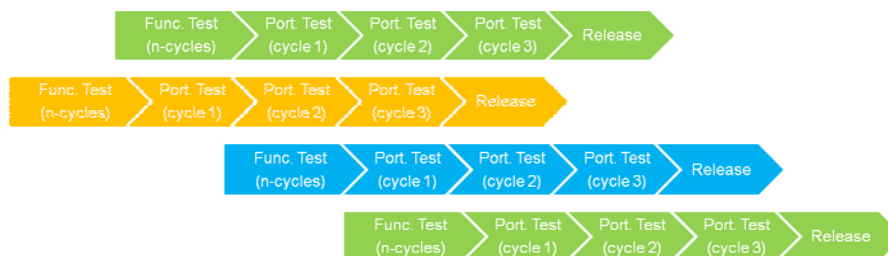
## Context

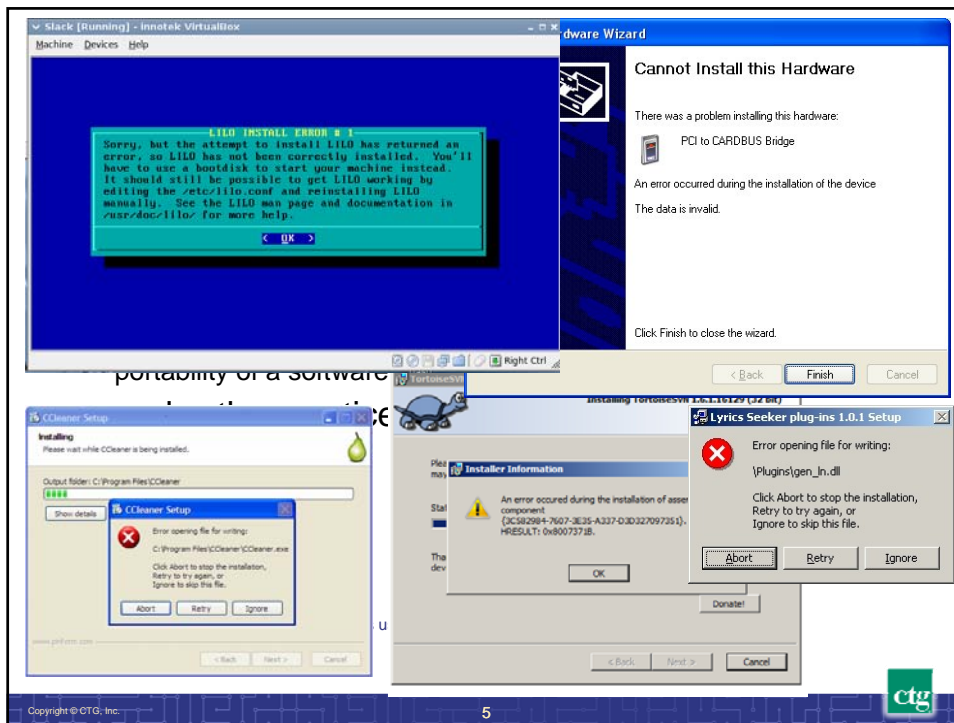
- Software centrally developed but locally adapted
  - Depending on regulations
  - Increased complexity due to multiple configurations



## Context

- Impact on testing
  - Functional testing needed on different levels
  - Portability testing of configurations
  - Country release run in parallel and independent of each other





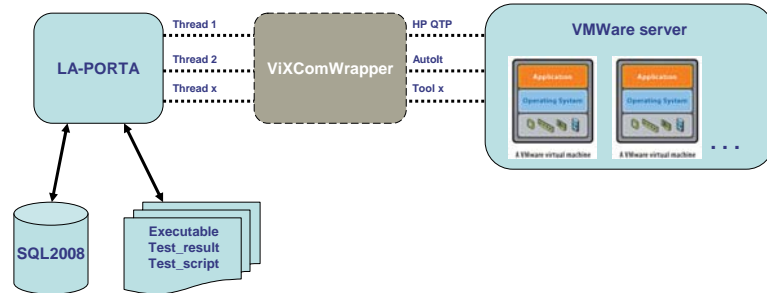
## Portability testing

- Portability testing is often “forgotten”
  - Physical resources/test environments
  - Time intensive – repetitive task (-> regression testing)
  - Grey zone (application development -> infrastructure)
  - No clear goals defined

Is it possible to automate portability testing using virtual environments?

## Solution

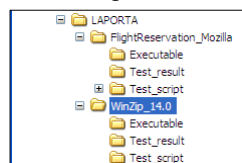
### ▪ Launch Automated Portability Testing



## Solution

### ▪ LA-PORTA

- Lean & mean as possible
- Concept of package (= AUT)
- 3 main parts
  - Configuration  
= parameters – VMWare related – packages
  - Compose test run  
= link package to VMImage – test script – pre/post conditions
  - Manage test set  
= launch test run in parallel threads
- Recurring folder structure per package



## Solution

### VMWare Server<sup>1</sup>

The screenshot displays the VMware Infrastructure Web Access interface. The main content area shows the configuration for a virtual machine with the following details:

- General:** Hostname: 330WY33.ctg.com
- Processors:** Intel(R) Core(TM)2 Duo CPU P8400 @ 2.26GHz, 1 CPU x 2 Cores, Usage: 282.00 MHz
- Memory:** 3.45 GB, Usage: 2698 MB
- Datstores:** A table showing standard and VMfiles datstores with their capacities and free space.
- Networks:** A table showing network configurations for Bridged, HostOnly, and NAT.

Name	Capacity	Free Space	Location
standard	109.04 GB	26.83 GB	D:\Virtual Machines\
VMfiles	109.04 GB	36.83 GB	D:\Documents and Settings\ademey\My Documents\VMfiles

Name	VMnet	Type
Bridged	vmnet0	bridged
HostOnly	vmnet1	hostonly
NAT	vmnet8	nat

<sup>1</sup> - Business value of virtualization (Forrester report) on <http://www.vmware.com/files/pdf/solutions/Business-Value-Virtualization.pdf>

## Solution

### VixComWrapper

- Wraps VM VixCOM-API (original C# library)
- Enables automating VMWare tasks
- Synchronous implementation of functions
  - No need for callbacks – polling/blocking
  - Multitasking can be done in threading
- Smooth integration in Visual Studio
  - Full IntelliSense support
  - Throw exceptions in case of error (not only error code)

## Solution

- Test tools
  - Any tool could be used depending on goal of portability testing  
=> test tool script is called from .vbs script
  - Autolt (<http://www.autoitscript.com/>)
    - Free ware BASIC-like scripting language automating Windows GUI
    - Quick start for writing basic scripts
    - Initially for automating tasks during PC roll out situations
    - Scripts could be compiled to stand-alone executables
    - Object spy like functionality
    - Script editor (but no debug functionality!)
    - Good manual

## Demo

- Time for demo...

## Conclusions

- Portability testing is important but often forgotten
- Hidden area for added value of test automation
- No need for big/expensive tools to start
- Available API's facilitates integration in own solution

**Thank You!**

Questions/Comments

[wim.demey@ctg.com](mailto:wim.demey@ctg.com)