

[Presentation](#)

[Bio](#)

[Return to Main Menu](#)

P R E S E N T A T I O N

T6

Thursday, Dec 7, 2000

Test Environment Management

Steve Littlejohn

Test Environment Management

A Forgotten Basic

Steve Littlejohn

steve@simgroup.co.uk

Agenda

- ***Test Environment Management Disasters***
- ***Test Environments and the brave new World***
- ***Configuration Management - needs and wants***
- ***Test Environments vs Data Requirements***
- ***Test Environment Management - gaining control***
- ***Implementing Test Environment Management***
- ***Summary***

**Test Environment Management -
Disasters, an illustration of what can
happen....**

➤ *Why Disaster?*

- A good environment is a fundamental requirement
 - ◆ For most any old environment will do
 - ◆ Developers do care but put up with what is available
- Legacy systems on Mainframe always difficult
 - ◆ Tend to have been grown organically with applications development
 - ◆ Application integration and development “too involved and slow”
 - ◆ Environment maintenance ignored except by necessity
- Unix, eCommerce - requirements “bolted on”
 - ◆ Environments available until live then disappear
 - ◆ Test Environments generally do not match target

➤ **Example 1 - Legacy System, Mainframe**

- 3 Test Environments
 - ◆ Development/Unit Test
 - ◆ System Test
 - ◆ User Acceptance
- No version control or configuration management
- Testing normally tried and abandoned after Unit Test
- Batch processes run and abandoned if through program
- User Acceptance environment only used by some
- Production was the main test environment

➤ *Example 2 - Unix*

- Machines bought, network set up, according to size requirements
- Package installed, configured for use
- Customisation takes place
- Testing becomes an issue, demands environments
- Training becomes an issue, demands environments
- Sized boxes not enough to meet demands
- Environment sharing becomes a necessity and an issue
- Performance, Load and Stress squeezed into last week
 - ◆ Many faults found with machine configuration

➤ **Example 3 - eCommerce**

- Requirement is to integrate to current platforms and data
- Unix box sized and purchased
- Application development partly outsourced
- Security becomes an issue, cannot be tested due to environment
- Performance, Load and Stress not seen as a requirement
- Testing becomes nightmare
 - ◆ Integration to shared test environments reduces time available
 - ◆ Data available not good enough for application
- Final realisation that different box required for live environment

People tend to overly concentrate on hardware and infrastructure as opposed to configuration management and data.

➤ **General Test Environment Mistakes**

- No recognition of test environments importance
- If recognised, no true assessment of the test requirements
 - ◆ Performance, Load and Stress testing is the prime example
 - ◆ “We bought way over size so performance will not be an issue”
 - ◆ “We can use the test box and extrapolate the results for live”
- Poor use of version control, both source and environment objects
- Configuration management piecemeal, if any
 - ◆ Source, Compatibility, Operating systems, Configuration parameters
 - ◆ Data - meta, static, input, parameters, sources of, integration
 - ◆ Files, Databases, Connectivity, Network, Constraints

➤ *Test Environment Issues*

- Test Environments are an unrecognised neglected asset
 - ◆ Poor use made for testing, ignored until needed, quality suffers
 - ◆ Development teams incur cost to sort out, project delays
- Become difficult to maintain or allowed to become out of date
 - ◆ Tests fail due to hidden issues, or date 6 years ago, quality suspect
 - ◆ Uncontrollable development bug analysis cost, project delays
- Much time wasted in attempting to meet test requirements
 - ◆ Data unknown, new data added, data items “fudged” , quality suspect
 - ◆ Costs rise quickly, requirements not met, many bugs and delays

➤ *Test Environment Issues*

- Package data integration not always known
 - ◆ Lack of knowledge leads to poor test requirements, quality poor
 - ◆ Package customisation error rise, increased development cost
- Sizing only recognises production needs, testing an afterthought
 - ◆ Test environments shared, tests and time impacted, quality suffers
 - ◆ Testing “squeezed” to meet deadlines, heavy cost of Live bugs
- Performance, Load and Stress not considered important
 - ◆ Hardware test requirements ignored as well Software, quality ????
 - ◆ If all configured correctly....timebomb ticking.....Cost of Failure???

➤ *Test Environment Issues*

- Small sized environments restrict tests possible
 - ◆ Not always true, normally filled with previous data, quality suspect
 - ◆ Best use not made, much data development at extra cost, delays
- Internal or external system connectivity not available
 - ◆ Tests “checked” by file formats or similar, quality suspect
 - ◆ Problems arise, extra development costs occur, many Live bugs
- Automation fails most often due to the test environment
 - ◆ Automation use restricted, becomes niche tool, quality poor
 - ◆ Expensive tools used by a few, shelfware, high license fees

Test Environments - the Brave New World

➤ *Downsizing Pains while Growing*

- Non Mainframe platforms “Cheaper, Better, Faster”
- Unix/PCs - Don't need Sys Progs, DBAs, Analysts, Programmers
 - ◆ Need other specialists - Unix, Network, PCs, DBAs!!!
- Standards not required -
 - ◆ “It's a PC - standards don't apply”
 - ◆ “It's a Unix box - we don't need that stuff”
- Project Lifecycles changed
 - ◆ Waterfall, CASE, RAD - Methodologies became fashionable!!!
- **All ignored testing and environments!!!!**

➤ *The Rise of the Internet and eCommerce*

- Early Websites became advertisement for Companies
- Basic information and contacts details, not integrated
- Normally single Server to keep isolated
- Next level gave more information
 - ◆ Integrated to some systems - product and price info
 - ◆ Fill in order form and receive goods and invoice
- Now we have eCommerce...
 - ◆ Fully integrated systems with up to date information
 - ◆ Credit card transaction, credit checking facilities
 - ◆ Multi-Server system, 24 hour, 7 days a week availability

➤ *Test Environments during this time...*

- Easy in early days...
 - ◆ Simple Server, run the tests using the limited data required
 - ◆ Security, Performance and Availability not seen as issues
- Next generation
 - ◆ Security becomes the issue, environments difficult for testing
 - ◆ Performance testing raised in profile, again difficult to test
- eCommerce requirements...
 - ◆ Environment with internal and external interfaces
 - ◆ Multi-Servers required, ability to test as Developer and Client
 - ◆ Security and Performance test capability a must

➤ *The Future (is already here!!!)*

- WAP Technology
 - ◆ Systems already developed
 - ◆ Test Environment is a mobile phone!!
- Palmtops
 - ◆ Requirement for more systems to interface
 - ◆ Web enabled and Physical linkages required
 - ◆ Test environment is a hand held box!!
- Next hardware technologies - watches, wrist commands etc..
- Test environment management principles still apply
- The same mistakes will be made!!

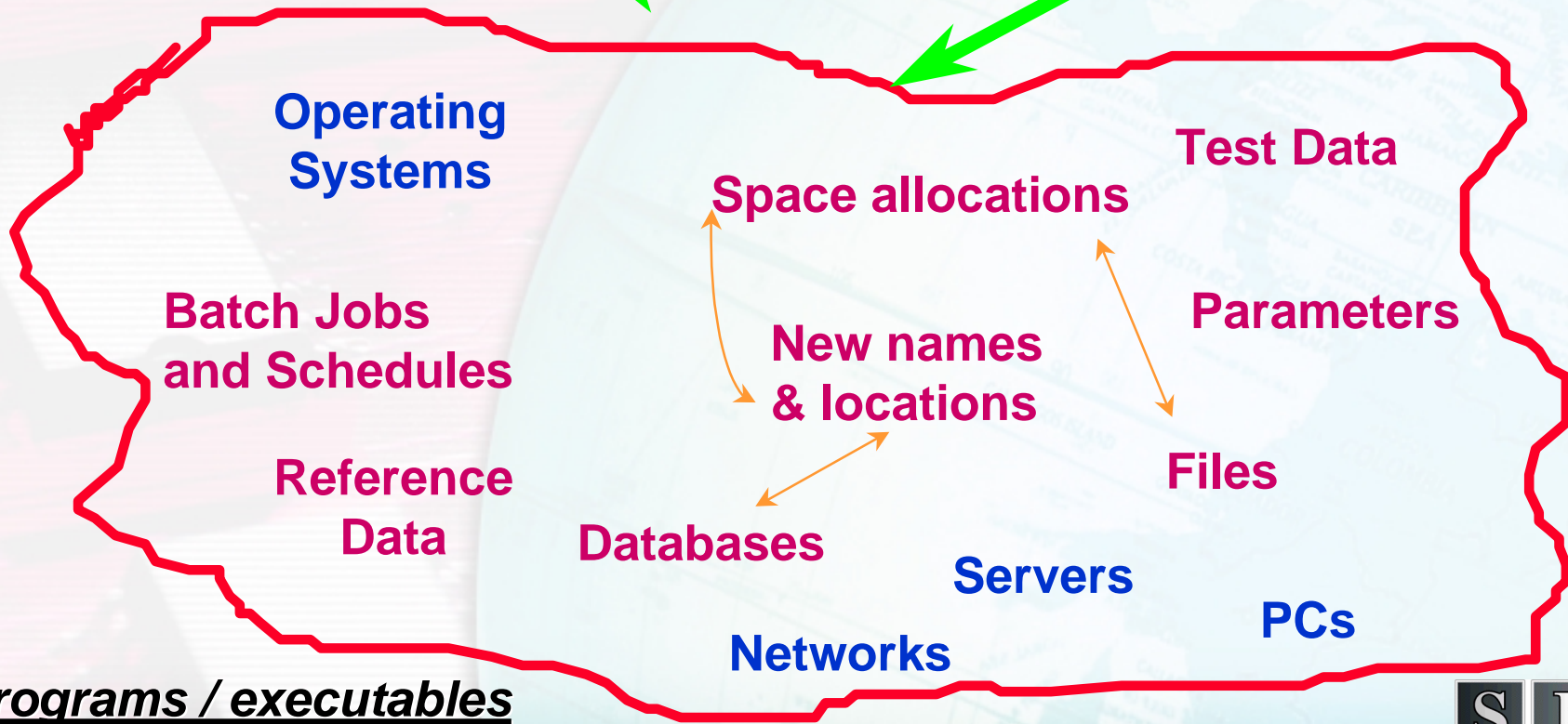
Test Environments - Configuration Management Needs and Wants

Test Environment Management - Environment Contents

GRESHAM

Isolation from development
and live environments

Problem, change
& release CONTROL



➤ *Test Environment Creation*

- Much more technical exercise on today's Client/Server systems
 - ◆ Many more factors to control, configure and record
- Setting up of Production environment copy not a necessity
 - ◆ Number of Servers reduced, Hard drive capacity reduced
 - ◆ Limited PCs available, "standard" PCs or found in corner
 - ◆ No "clean" environment with Production desktop configuration
 - ◆ Performance testing always an issue in today's systems
- Configuration recorded?
 - ◆ Not always, if an element fails, could be no method of reproduction
 - ◆ Management systems not seen as a necessity

➤ **Configuration Management**

- Normally stops at the Executable objects
 - ◆ Version Control system available
- Each department then has a method of recording configuration
 - ◆ Networks
 - ◆ Databases
 - ◆ PC Support
 - ◆ Unix Support
- Configuration recorded?
 - ◆ Data requirements tend to be ignored
 - ◆ No overall management or control, departments independent

➤ *Management Requirements*

- Overall Configuration Management system
 - ◆ Provide the ability to reproduce the complete environment
 - ◆ Improved ability to understand the environment
 - ◆ Improved ability to troubleshoot environment problems
 - ◆ Centrally recorded, available to all
- Importance of Configuration understood
 - ◆ Better basis for obtaining correct test environments
 - ◆ Test Requirements more able to be met
 - ◆ Test Kit recognised as needed current rather than old
- Impact of change can be readily identified

Test Environments - Data Requirements

➤ *Test Environment Data Requirements*

- High Coverage Test data
 - ◆ Base Test Data
 - ◆ Specific Test Conditions
- Manageable Volumes of data
- Responsive to project testing time frames
- Able to support each level of testing
- Ability to back-up/restore
- Data must have integrity
- Internal and External data interfaces

➤ *Data Requirements for Test Environments*

- High Coverage of test conditions
- Need to know data vs. condition
- Need data matched to the testing need
- Ability to vary quantity to the testing need
- Ability to back-up/restore
- Ability to refresh when required
- Able to set dates for particular run scenarios

➤ *More on Test Data....*

Unit testing

➔ *Real test cases*

System testing

➔ *Integrity of data*

UAT

➔ *Real business scenarios*

Pre-Production testing

➔ *Real environments*

Performance testing

➔ *Real volumes*

- **Full copies of Production unmanageable**
- **Difficult to add or maintain each time**
- **Meeting these needs is where management is applied**

Test Environment Management - Gaining Control

➤ *Gain Control of what?*

- Gain understanding first
 - ◆ Test Environments vs. Production environment
 - ◆ **Configuration - Quantify differences, which are important?**
 - ◆ **Interfaces available and required-**
 - ◆ Data
 - ◆ **Purpose, Content and volume -**
 - ◆ **Sources**
 - ◆ **Integration**
 - ◆ Applications
 - ◆ **Objects associated**
 - ◆ **Relational integrity rules - Data or Application?**

➤ ***Gaining Control***

- Determine boundaries of applications
- Determine the data flow through applications
- Understand how the application works

- **DOCUMENT ALL YOU HAVE FOUND!!!!**

Test Environment Management - Implementation

➤ *So what is missing?*

- Test Environment Management is a new role, part of test culture
- The culture should ensure the environments are
 - ◆ Refreshable
 - ◆ Maintainable
 - ◆ Up to date
 - ◆ Flexible
 - ◆ Integrated
- Data requirements must be able to be met
- Configuration management is a must for all objects
- **Project Team time on environments is too costly**

➤ *First Steps, Environment basics*

- Ensure all databases and files are latest configuration
- Remove all data that is superfluous
- Refresh Reference and Static data
- Add interfaces required
- Test the environment, back-up/restore procedures
- Automate the processes for the above wherever possible
- Seek a Project requiring an environment
- Source the required data for the testing needs
- Deliver the environment
- This becomes your test for best process

➤ ***Implement Best Process***

- For Data Refresh and Maintenance
- For Data Identification for Test Requirements
- To build the test environment
- To provide data volume scalability
- For Environment Maintenance
- To provide estimates, and fit into Project Lifecycle

➤ ***Automate Processes wherever possible***

- Provides an easy method to build, populate and refresh
- Gives greater control to manage environments

Summary



- *New platforms have not improved Test Environment use, control and management*
- *The importance of the Test Environment has been largely overlooked in the race to develop*
- *Implementation of Automation fails most often due to poor or shared test environments*
- *Configuration Management is necessary for all objects associated with a Test Environment*
- *Data content and integration must be known to fulfil the test requirements*
- *Test Environment Management frees the Project Team, makes assets reusable, reduces costs*



Thursday 7 December 2000

T6

Test Environment Management

Steve Littlejohn

Steve Littlejohn has been involved in the IT industry for about 20 years. He has worked in most IT roles from operations through to project management via programming, analysis and design before specialising in testing for the past eight years. With SIM Group he is a Senior Consultant and has worked on testing projects for a variety of clients within many industry sectors. He has a special interest in the techniques of test environment management and automation and their use with automated testing.

