

Automated Testing

Presentation to EPICS Collaboration Meeting
2007 @ DESY

Paul Gibbons PhD
Diamond Light Source Ltd.



Automated Testing – Why?

- Code Guru

- Edit
- Compile
- Link
- Publish

- No testing needed.

(Also turn off all compiler warnings as I never write incorrect code.)

Automated Testing – Why?

- We live in the real world



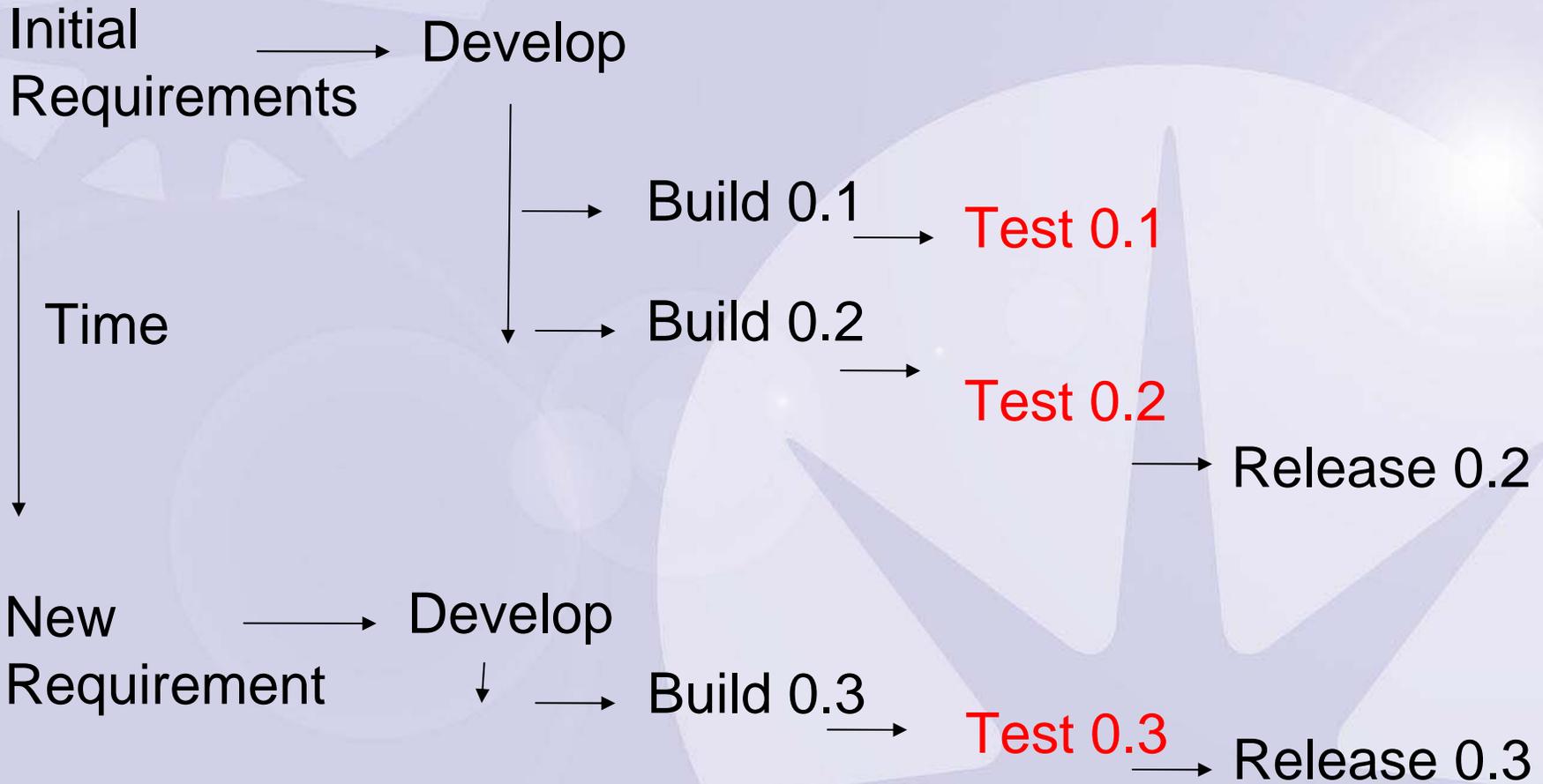
Automated Testing – Why?

- Others will need to change it.



- Can you trust them not to mess up?

Automated Testing – Why?



Automated Testing – Why?

- Manual testing is:
 - Expensive
 - Time consuming
 - Error prone.

Automated Testing – Why?

- Automated Testing:
 - Repeatable.
 - Running the test requires little human intervention.
 - Can encapsulate the requirements better than a requirements document – is always up to date.
 - Tests not only the code you know you have changed but also the code you think you have not.
 - Expensive up front investment.

Automated Testing – Why?

Late night call out to modify Fred's module.

- Make modification.
- Automated tests show no problems.
- Manually test new feature.
- Go home confident 😊.

Next day review change with Fred and add test to automated tests.

Automated Testing – Why?

- New requirement calls for re-factoring
 - Before adding any new feature you gradually refactor the code.
 - At each point run automated test.
 - At end of refactoring you have confidence that the software still behaves correctly and you are ready to add new features.

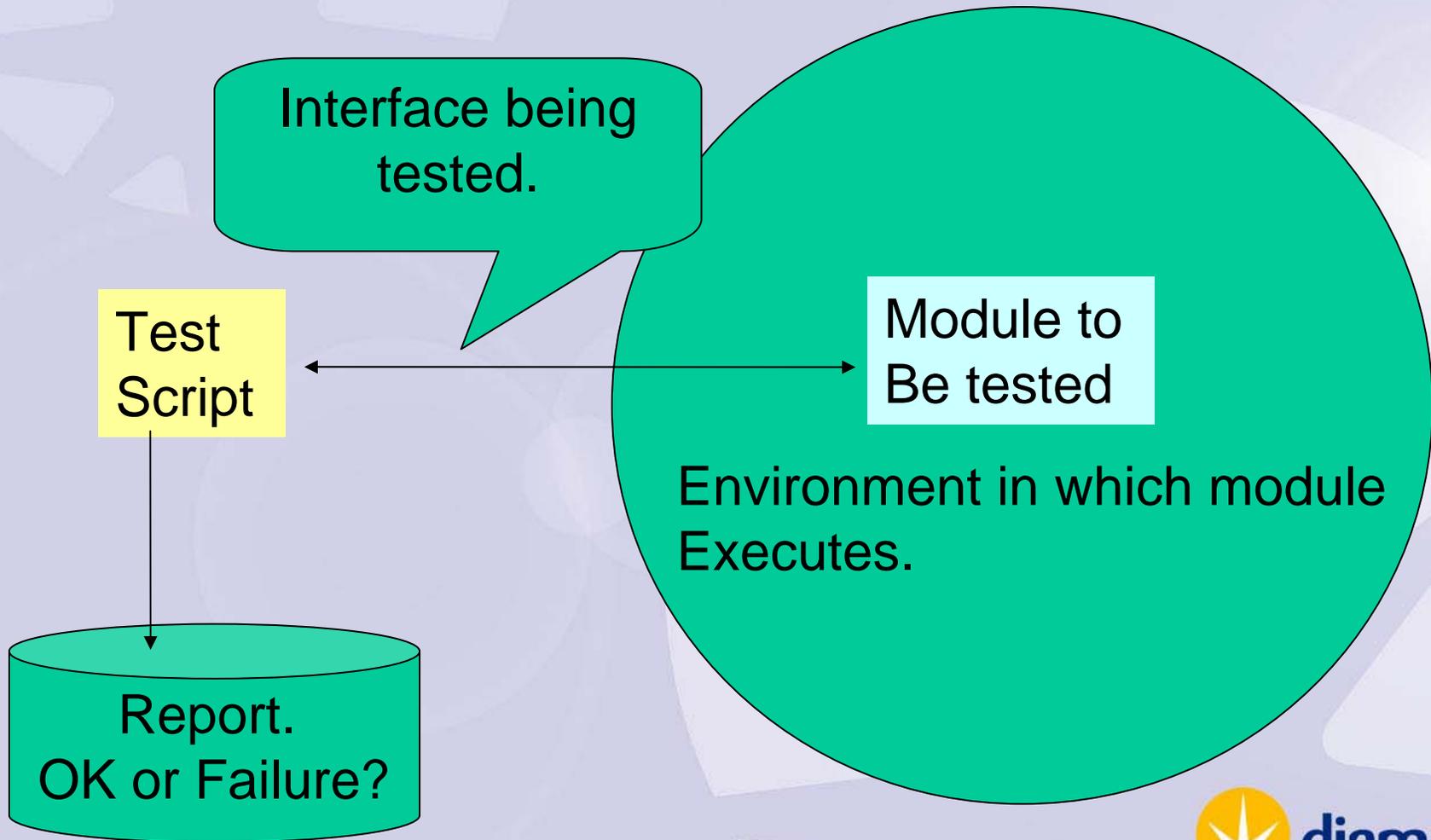
Automated Testing – Why?

- Testing on its own is not a panacea.
 - Different defect-detection techniques
 - Reviews, formal/informal design/code inspections, personal desk checking, prototyping, unit test, integration test, system test, beta tests
 - Defect Detection Rates:
 - None above 75%
 - Testing gives 30-40%
 - A combination is required to get > 95%.

Automated Testing – Why?

- Many other advantages of thinking about testing up front.
- See books on Test Driven Development.

Automated Testing – How



Automated Testing – EPICS

Types of IOC software at Diamond.

VME Register Interface

Serial Device Interface

Device Independent Records – e.g. Motor

Gensub records with
logic built into C function

Sequences

Frameworks: asyn, stream

Groups of ‘standard’ records that
together provide functionality
– aperture slits, diffractometer table.

Automated Testing – EPICS

Current testing is 99% manual.

Normally requires a complete system with hardware.

Automated Testing - EPICS

Option 1. Integration Test.

- Create a complete test system that includes hardware.
 - The expense precludes this for all software.
 - Testing is only possible once hardware is available

Option 2. Isolation

- Requires simulation of environment.

Automated Testing - EPICS

You want to test the logic of code.

Use the language of the interface.

A test environment should :

- not impose any constraints on the interface
- be simple and quick to construct.

Automated Testing – IOC

Logical Units that interface via records:

- Accompanying Test IOC
- Channel access script to perform the tests and generate a report.

What about the environment it requires?

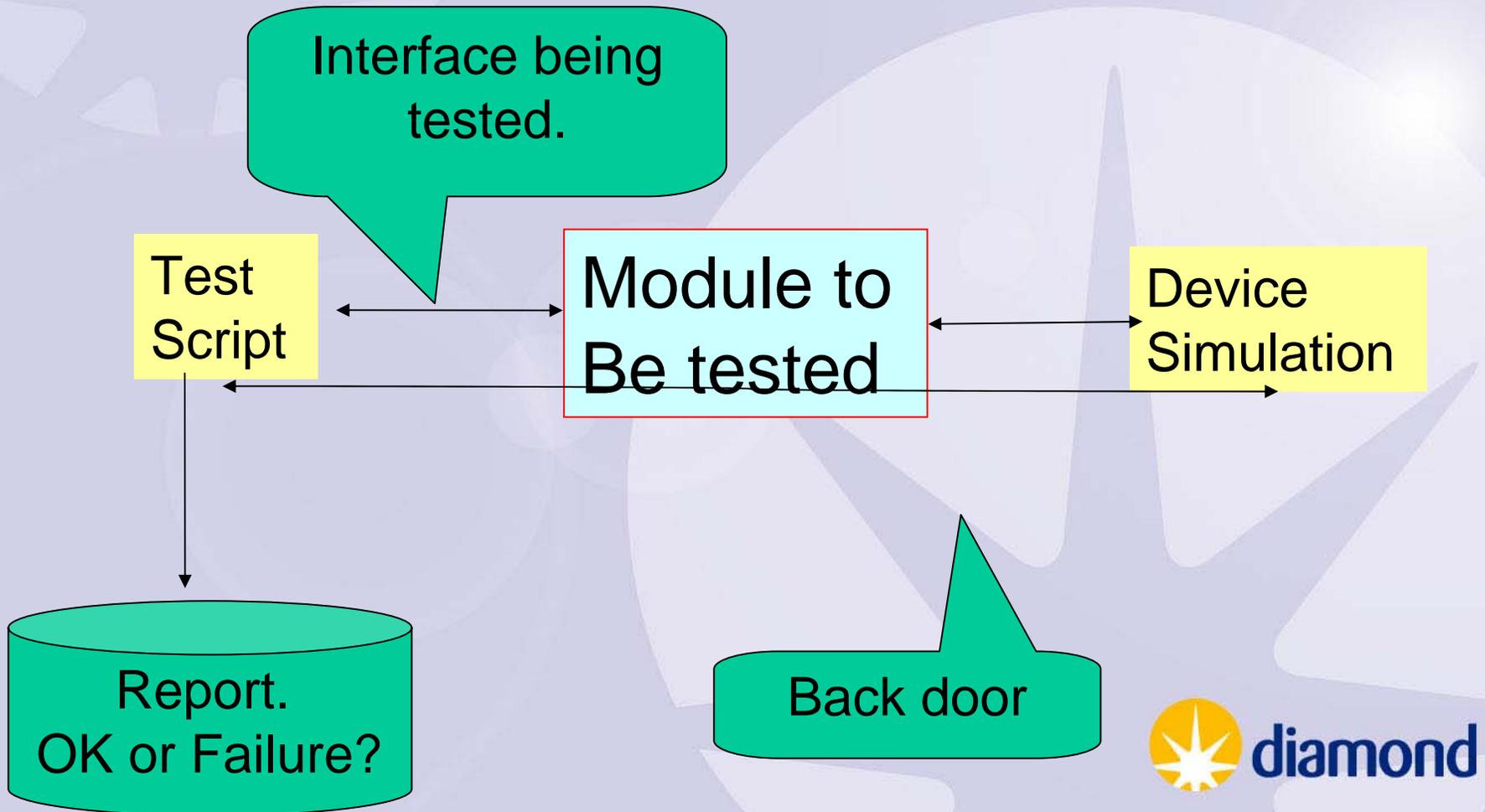
Simulation mode to be used in Test IOCs of other modules.

Automated Testing – IOC

C functions that can be built for Linux e.g. genSub systems:

Use CPPUnit

Automated Testing – Device Simulation



Automated Testing – Device Simulation

Record Simulation Mode

- Hardware loopback

Serial(RS232/IP) -

- asyn IP client
- Java – COSYLAB
- asynPort that runs Python

VME Register – simulation device drivers.

Automated Testing – Plan

- Many options. Need to agreed on standards for:
 - Test language – Should match interface being tested. For Channel Access use Python or Jython.
 - How to build the test.
 - How to run the test.
 - Reporting success and failure. xUnit XML format
Provide parsers to convert from other formats

Current work:

PyUnit + serial device simulation + use of
record simulation mode



Automated Testing - References

- Code Complete, Second Edition by Steve McConnell. ISBN-10: 0-7356-1967-0
- Various work by Kent Beck a founder of the Agile Manifesto.