

Regression Testing



January 2001

Topics

- ✓ Regression Testing
 - what is it?
 - why is it necessary?
 - traps and pitfalls
- ✓ Integration testing using regression testing
- ✓ Automation of regression testing

January 2001

Page 2

What is Regression Testing?

- ✓ Rerunning a predefined set of tests to ensure functionality that used to work still works
- ✓ Done after a change to the environment, the application or an interface



January 2001

Page 3

Benefits of Regression Testing?



- ✓ Increases chances of finding bugs caused by changes to the application
- ✓ Detects undesirable side effects caused by changing the operating environment
- ✓ Provides a 'new' way to do integration testing

January 2001

Page 4

After a Change You should Ask:



- ✓ Did the change work?
- ✓ Did it break anything else?
- ✓ Are the regression test cases still okay?

January 2001

Page 5

Traps and Pitfalls

- ✓ Adequacy of the tests is unknown
- ✓ Can't capture some results/actions
- ✓ Too many regression tests to run after each change
- ✓ Unrealistic expectations



January 2001

Page 6

Traps and Pitfalls



- ✓ No correlation of individual tests to functional or structural parts of application
- ✓ Changes to the system cause too many tests to fail
- ✓ Regression tests are out of step with the application

January 2001

Page 7

Tips



- ✓ Keep the tests simple
 - ignore all the outputs other than the one you are testing
- ✓ Keep the tests independent
 - pick and choose the ones you want/need to run

January 2001

Page 8

Tips

- ✓ Each test should start and end at a common point
- ✓ Structure scripts like good code
- ✓ Lock critical tests
- ✓ Package the state of the world with the regression test cases



January 2001

Page 9

Really Big Tip

- ✓ Separate the processes of regression testing for environmental changes from testing for application changes



January 2001

Page 10

Integration Testing using Regression Testing



Integration Rule #1

- ✓ Do a walkthrough of the interface requirements
- ✓ Check for:
 - Number of parameters
 - Required vs. optional
 - Order of parameters
 - Type / data format
 - Length
 - Definition



Integration Rule #2

- ✓ Use systematic incremental integration
 - integrate one piece at a time and test that everything works AND still works each time a new piece is added



January 2001

Page 13

Integration Testing using Regression Testing Technique

1. Start with something that will run standalone
2. Create and execute test cases to exercise functionality
3. Add next piece



January 2001

Page 14

Integration Testing using Regression Testing Technique



4. As each piece is added, use tests to check:

- the new piece works alone
- the new piece performs its function when connected to established working pieces
- adding the pieces didn't break something else

January 2001

Page 15

Automation of Regression Testing



Should we automate?

- ✓ It is NO silver bullet
- ✓ Will require structured programming knowledge
- ✓ Like all testing, will eat up time
- ✓ Tools make it very easy to create a lot of action but they do not make it easier to find the serious bugs

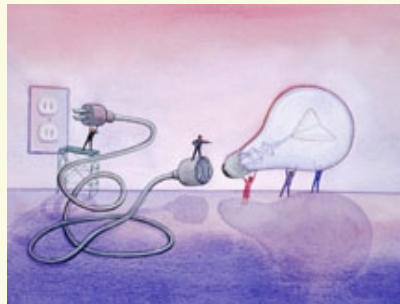


January 2001

Page 17

But will it save time?

- ✓ It will make the current project slower
- ✓ Time savings will begin to show up on the next project
- ✓ Scripts will have to be updated to include new/changed functionality



January 2001

Page 18

Remember



Remember



- ✓ Regression tests are documented and repeatable
- ✓ Each time application is changed, regression tests must be updated to reflect change
- ✓ Perform regression testing after a change to confirm nothing has been broken

Also Remember

- ✓ Structure integration test to use regression tests
- ✓ Automation of regression tests will start to show time saving benefits in second and subsequent releases



January 2001

Page 21