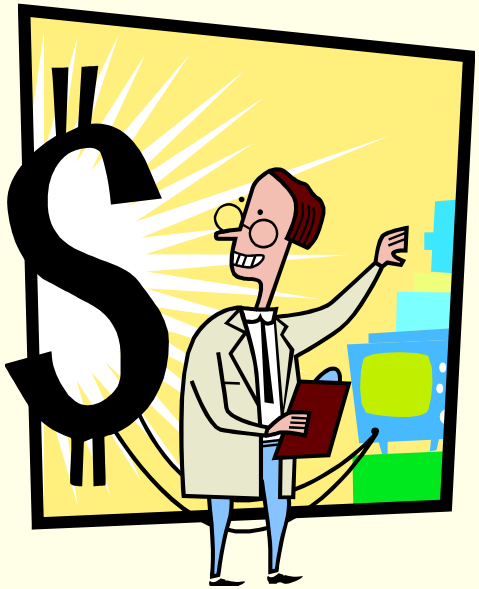
# … and after unit testing …

Integration Testing, User Interface Testing, Validation, System Testing, …

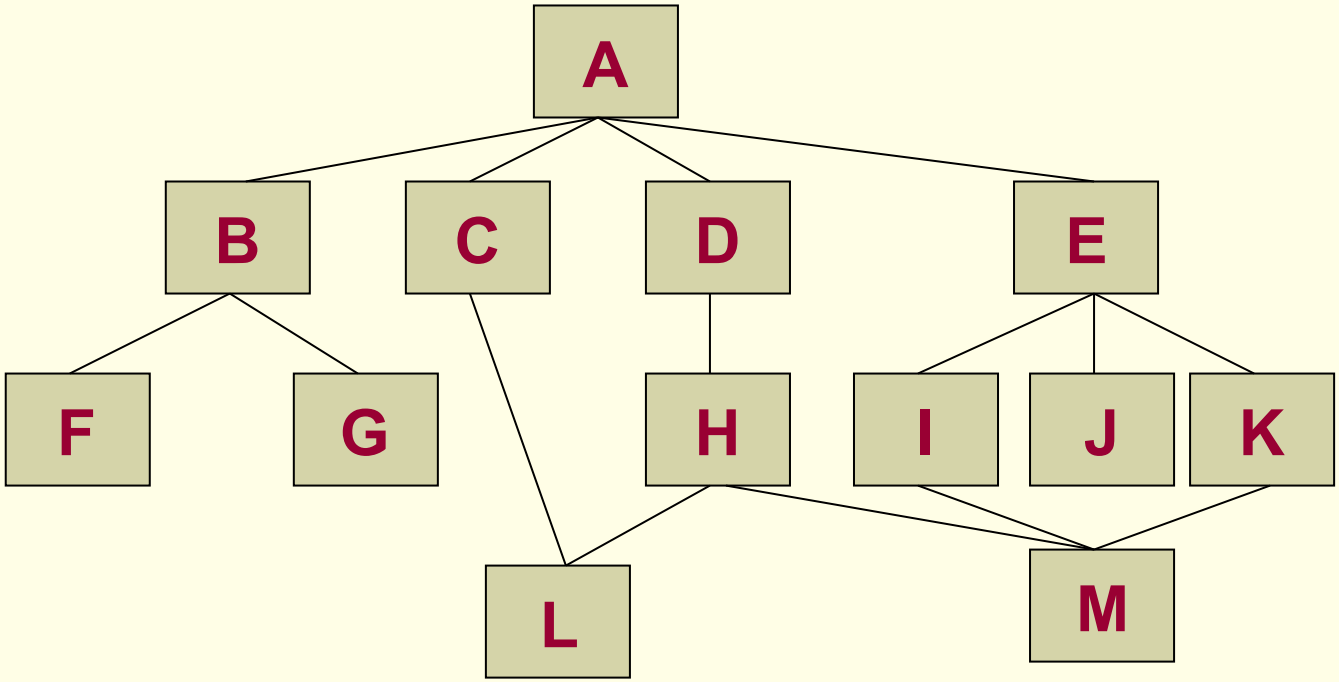
# *Testing, testing, …*

- Integration Testing
- User Interface Testing
- Validation
- System Testing



# Integration Testing

- Each of the following modules, shown below in the application's control flow chart, has finally passed unit testing standards. How do you plan to conduct integration testing?



# Integration Methods

- Top-Down
  - using stubs
- Bottom-Up
  - using drivers
- Depth-First
- Sandwich

***Regression Testing***

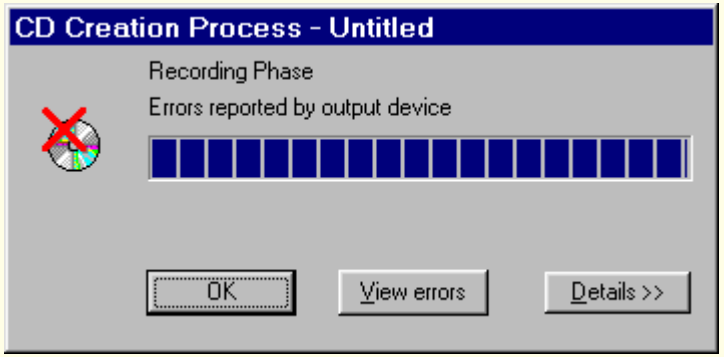
# User Interface Testing

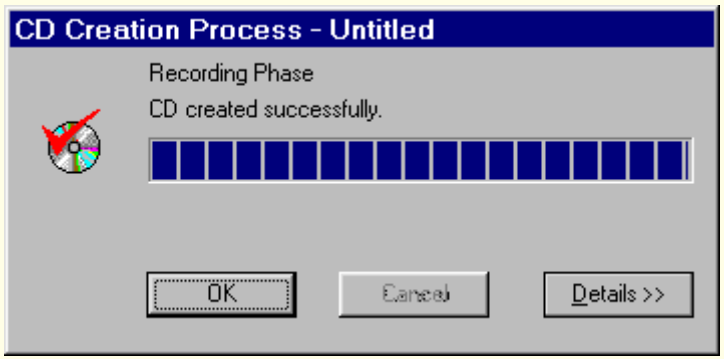
- **User Interface Validation**
  - paper prototypes made early are great
- **User Interface Evaluation**
  - Expert / Heuristic Evaluation
  - Label Testing
    - is a "rose" still a rose by any other name?

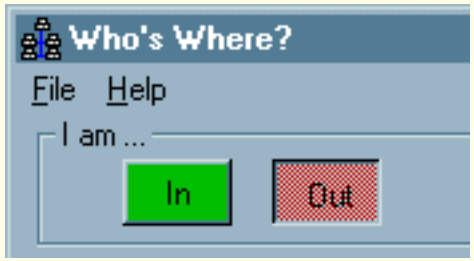
# User Interface Testing

- **User Interface Testing**
  - Cognitive Walkthroughs
  - Usability Testing
    - hopefully the SRS specifically defines usability criteria
  - Random Testing
    - automated random key/mouse presses

# What's the problem?

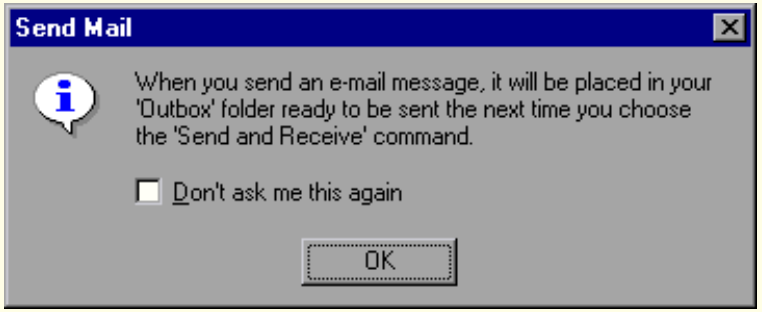



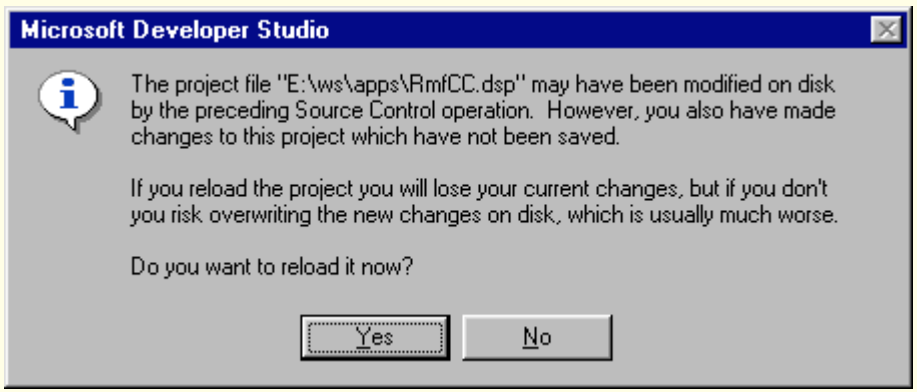




http://homepage.mac.com/bradster/iarchitect/color.htm

# Dialog Boxes







# Cognitive Walkthroughs

- **Participants**
  - real end user - sits at keyboard and performs tasks
  - evaluator - takes notes and asks questions, mostly quiet
  - developer - probably hidden or watches video
- **Results**
  - effectiveness
  - Will the user associate the next action with the appropriate interface control?
  - Will the user notice an action is available?
  - If the correct action is performed, will the user see that progress is being made toward solution of the task?

# Heuristics

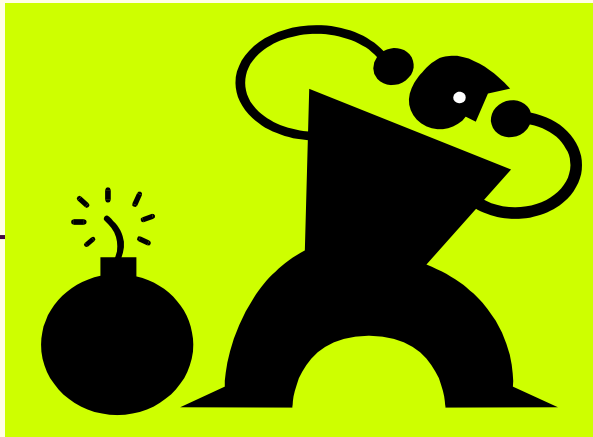
- Simple and natural dialogue
  - Aesthetic and minimalist design
- Speak the user's language
- Minimize user memory load
  - Recognition rather than recall
- Consistency
- Feedback
  - Visibility of system status
- User control and freedom
  - Clearly marked exits
  - Shortcuts and Flexibility
- Good error messages
  - Help users recognize, diagnose, and recover from errors

modified from http://en.wikipedia.org/wiki/Heuristic_evaluation

# Validation Testing

- Alpha Testing
- Beta Testing

# System Testing



- Recovery Testing
- Security Testing
  - use software to analyze source code for stack buffer overflow attacks, etc
- Stress Testing
- Performance Testing

# Object-Oriented Testing

- Why is OOT more difficult than regular testing?
  - inheritance of functions
  - inheritance of data
  - abstract classes
- Testing the Design is much more important because the code level is very difficult to test.
- Testing should be aimed at the "class" not the "module"
  - this is because the operation of a module probably depends on how its inherited

# *And the Moral of the Story is…*

Use Testing Tools

# And the glue that holds it all together…

- **The Test Plan**
  - who
  - what
  - when
  - where
  - how

# Test Plan Considerations

- What are the critical or most complex modules?
  - make sure they get integration tested first
  - probably deserve white-box attention
- Where have you had problems in the past?
- Third-Party delivered components?
- What training is required?
  - conducting formal reviews
  - use of testing tools
  - defect report logging

# IEEE 829 - Standard for Software Test Documentation

Recommends 8 types of testing documents:

1. Test Plan
   - *next slide*
2. Test Design Specification
   - expected results, pass criteria, ...
3. Test Case Specification
   - test data for use in running the tests
4. Test Procedure Specification
   - how to run each test
5. Test Item Transmittal Report
   - reporting on when components have progressed from one stage of testing to the next
6. Test Log
7. Test Incident Report
   - for any test that failed, the actual versus expected result
8. Test Summary Report
   - management report

http://en.wikipedia.org/wiki/IEEE_829

# Test Plan Contents (IEEE 829 format)

1. Test Plan Identifier
2. References
3. Introduction
4. Test Items
   *see next slide*
5. Software Risk Issues
6. Features to be Tested
7. Features not to be Tested
8. Approach
9. Item Pass/Fail Criteria
10. Suspension Criteria and Resumption Requirements
11. Test Deliverables
12. Remaining Test Tasks
13. Environmental Needs
14. Staffing and Training Needs
15. Responsibilities
16. Schedule
17. Planning Risks and Contingencies
18. Approvals
19. Glossary

http://en.wikipedia.org/wiki/Test_Plan

# 4. Test Items

- Requirements Specification
- Design
- Modules
- User/Operator Material
  - the user interface
  - User Guide
  - Operations Guide
- Features
  - response time, data accuracy, security, etc
- System Validation
  - alpha and beta testing

somewhat based on IEEE 829

# Reality Check

- ***When is more testing not cost effective?***