



# Teaching Software Testing

**Dennis J. Frailey**

**(Retired) Principal Fellow, Raytheon Company**

**Adjunct Professor of Computer Science and  
Engineering, SMU**

**Adjunct Professor of Computer Science and  
Engineering, UT Arlington**

[Frailey@ACM.ORG](mailto:Frailey@ACM.ORG)

[Frailey@Lyle.smu.edu](mailto:Frailey@Lyle.smu.edu)

[Dennis.Frailey@UTA.EDU](mailto:Dennis.Frailey@UTA.EDU)

**Presented at CSEET 2016**



# What Testing Techniques Are Used in Your Software Projects?

Most of our projects involve development of **new hardware** as well as software

Most require **high reliability** and some are **safety critical**



- *System level requirements* are just as important as software requirements
- We often require use of *special purpose testing equipment* that is unique to the product being developed
- Test *procedures and results must often be carefully recorded (documented)* and are often audited



# What Testing Techniques Do New Hires Not Know?

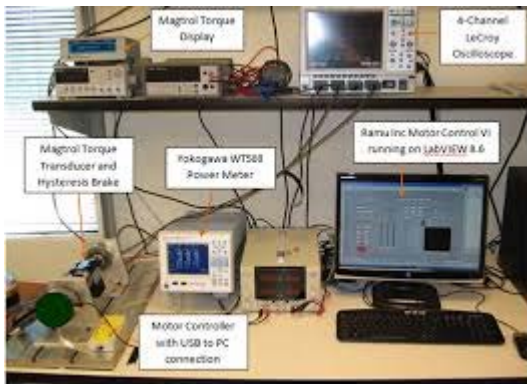
Results vary, but:

- Many know software but know *too little about hardware*
- Some don't seem to understand the *underlying principles* – only how to use specific languages or tools
- Few of them know *testing discipline*:
  - Developing effective test plans
  - Documenting the results of testing (test reports)
  - Testing the test code



# What Testing Tools Do You Use?

- **Product-specific Test Equipment and Tools**
  - Software staff must participate in design and definition of these tools





# What Testing Tools Do You Use?

- **Product-specific Test Equipment and Tools**
  - Software staff must participate in design and definition of these tools
- **Simulation**
  - Simulate the system before building it
  - Compare actual results with simulated results to find underlying causes of problems





# What Testing Tools Do You Use?

- **Product-specific Test Equipment and Tools**
  - Software staff must participate in design and definition of these tools
  
- **Simulation**
  - Simulate the system before building it
  - Compare actual results with simulated results to find underlying causes of problems
  
- **Automated Testing / Test Generation**
  - Requires very good discipline in defining requirements
  - Can potentially save a lot of time and money



# What Are Your Testing Needs Where Current Staff are Not Well Prepared and Research is Needed?

- **Integration of hardware and software testing methods and tools**
  - The problems are often complex and may involve both hardware and software failures
- **Automatic testing**
  - Sometimes the system is too high in performance to do traditional testing
  - Examples:
    - With high speed networking, you may need to see and track individual packets
    - With multi-core systems the interactions between cores can be very hard to see with traditional testing methods



# Bottom Line

- **You Must Consider the Complete System/Product, Not Just the Software**
  - Software developers must understand how the hardware works
  
- **Testing Begins At the Start of the Project**
  - Test plans should be initiated by system/software designers, not by “testers”.
  - “Test First” – don’t write code if you don’t know how you will test it
  - Requirements may change, but you must make them as specific as possible
    - Are the requirements testable?
    - Are they unambiguous?





# Questions?

