Computer Engineering

Meeting 3:
Computer Engineering Disasters

Developed by José R. de la Cruz

Computer Engineering Disasters

- Computer Engineering is all about the details!
- Computer Engineers must make sure that the system they design *behaves* as intended by using proper testing techniques.
- Failure to properly test a system can result in a disaster that endangers us.
- Always remember the first Canon of the Code of Ethics: *Hold paramount the safety, health, and welfare of the public*.

Computer Engineering Disasters

- This video created by The History Channel showcases software derived disasters

- Video link

Computer Engineering Disasters

- So, why was the Ariane-5 rocket destroyed?
- Again, Computer Engineering is all about the details!
- Could a proper test procedure have caught the error?
- Who is to blame?
- Discuss these topics with your instructor

Software Testing

- In order to make sure that the designed system *behaves* properly, Computer Engineers use testing protocols and techniques.
- Testing must be done at every phase of the system lifecycle.
- Testing aims to:
  - Produce a quality system
  - Make sure that all requirements are met
  - Assure that the system satisfies the needs of the client

Software Testing

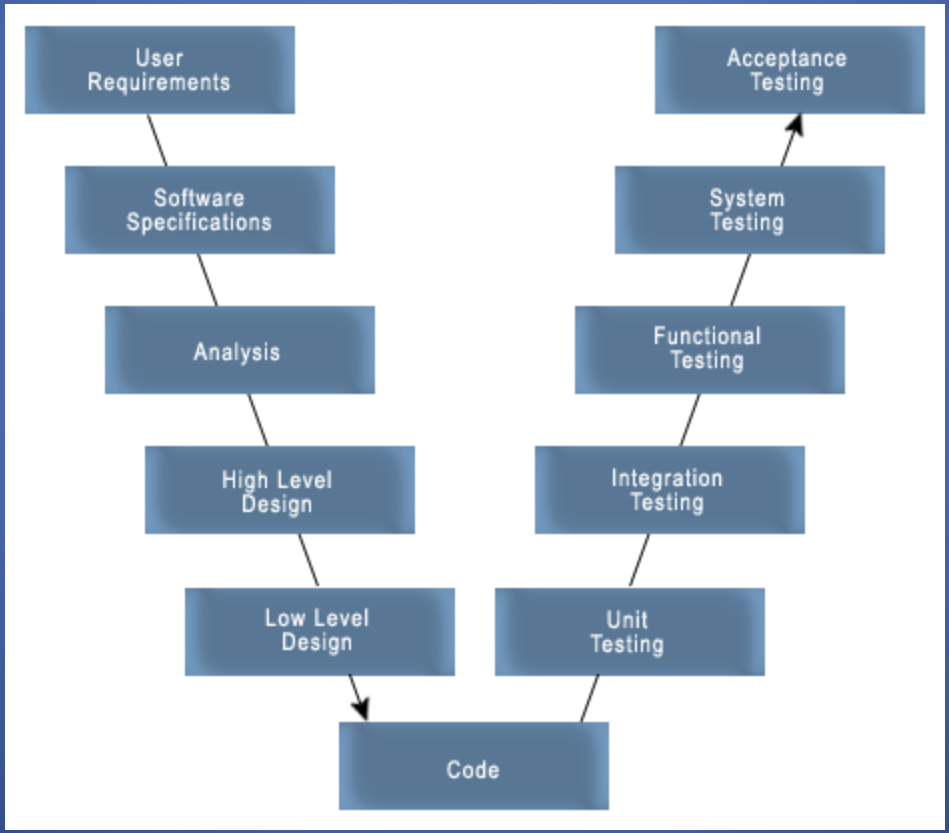
- There are three basic types of testing procedures:
  - black box
  - white box
  - grey box

Software Testing

- In the black box technique, the system's output is compared from the one expected from the input provided, but the tester does not have information about the details of how the system performs its duties.
- In the white box approach, the tester knows all the details of the system and watches how the components interact.
- In the grey box scenario the tester has limited knowledge about the inner-workings of the system.

Software Testing

- Software testing is performed at different levels.



Source: http://www.nivista.com/quality/quality.htm

Software Testing

- Think of a system as a collection of pieces.
- First, test all pieces independently (unit tests).
- Second, make sure that all pieces can work together (integration tests).
- Third, make sure that each activity that the system was designed for can be achieved by the system (functional).
- Fourth, ensure that the system works properly as a whole (system tests).
- Last, but very important, make sure that the customer can use the system at their facilities. This level requires system installation at the customer's premises.

Preventing Disasters

- Disasters can range from life-endangering to system malfunction.
- First, and foremost, make sure that you understand what the customer wants.
- Be keen to consider the environment in which the system will be installed (operating system, network facilities, etc).
- Consult with domain experts to make sure that all situations have been considered.
- Test, test, and then test some more
- Document all testing procedures and results
- And ALWAYS *hold paramount the safety, health, and welfare of the public.*

Preventing Disasters

- Make a plan which details how testing is to be carried out, and FOLLOW IT!
- Use flowcharts to test the basic concepts of the system (what happens if *X* fails?).
- When creating a timeline for system delivery make sure that enough time has been considered for proper testing.
- Remediate all incorrect situations at once (do not “do it for later”).

Conclusion

- Computer Engineering is all about the details!
- Even very small errors can result in a big disaster.
- Plan for testing.
- Test at every step.
- Your system must be reliable, on-time, and within budget.

Next Meeting

- Since this is a hybrid course, our next meeting will be online.
- A learning module will be available in Blackboard
- The content relates to your project.
- You should continue the project development with your group.

Questions