BCS3323 – Software Testing and Maintenance

Overview of Testing

Editors
Prof. Dr. Kamal Z. Zamli
Dr. AbdulRahman A. Alsewari
Faculty of Computer Systems & Software Engineering
alswari@ump.edu.my

Chapter Description

- Aims
  - To determine the necessary of the testing event.
  - Differentiate between the error, defect, and failure.
  - Identify the error, bug and failure creators.
- Expected Outcomes
  - Students can explain the purpose of the testing
  - Show some examples to support their understanding
- References
  - ISTQB
  - MSTB/GTB
  - http://www.softwaretestingclass.com/software-testing-tools-list/
  - http://www.softwaretestinggenius.com/articalDetails.php?qry=572#commentsList
  - https://www.guru99.com/software-testing-seven-principles.html

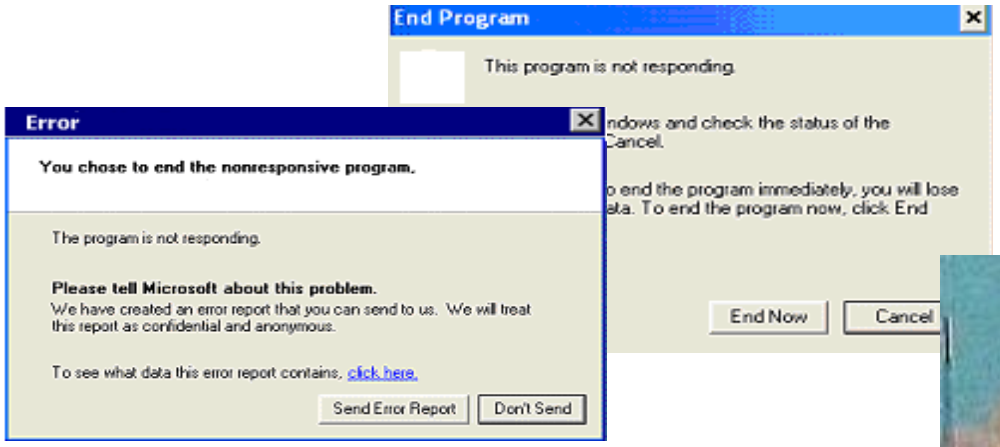
Software is everywhere

- Software is part of our lives.
- We use software everywhere i.e. our mobile phone, washing machine, air bag controller etc.
- Why do we opt for software rather than hardware whenever possible?
    - Easy to customize
    - Does not wear out
- Our dependencies on software raise issues on quality and reliability.

In God We Trust, The Rest We Test !!!

The need for software testing…

- **Software failures can lead to disastrous consequences**
  - Loss of data …..



- Loss of fortune …..

- Loss of lives …

Why do faults occur in software?

- Software designed and developed by human beings
  - Who has limitations of knowledge, not everything
  - Who have skills, but isn't perfect
  - Who has the potential to create mistakes (errors)
- When trying to deliver the software under increasing pressure to strict deadlines
  - No enough time to check.
  - The assumptions may be wrong
  - Systems may be incomplete

What do software faults cost?

- Lost huge cost such as in:
  - $ 7billion has been lost by Ariane 5
- Insufficient or nothing at all
  - minor tiresomeness
  - no visible or physical harmful impact
- software is not “linear”:
  - Some times small input value may causes a very large effect

Safety-critical systems

- Death or injury can be caused based on Faults in software.
  - Medical machine kills patients when using that machine to take radiation treatment (Therac-25)
  - Many trains, vehicle kill people based on failure in break systems.
  - Many aircraft crashes based on system failures
  - Suicide actions based on bank systems overdraft letters

why is testing necessary?

- because software is likely to have faults ✓
- to learn about the reliability of the software ✓
- to fill the time between delivery of the software and the release date ✗
- to prove that the software has no faults ✗
- because testing is included in the project plan ✗
- because failures can be very expensive ✓
- to avoid being sued by customers ✓
- to stay in business ✓

Testing Terminologies

Software Testing - definition

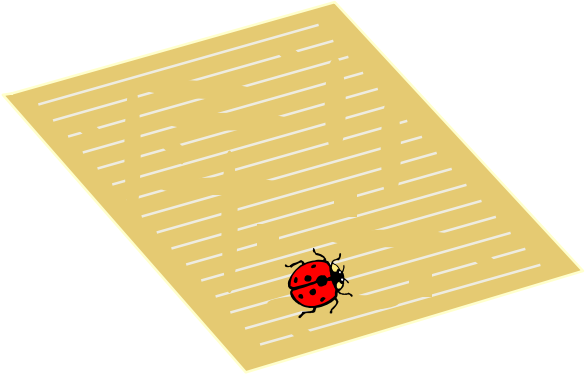
- Software testing is the process of **finding errors** (i.e. sometimes involves **executing the software** of interest) and of **validating the software**/system **against its specification. (**ISTQB**)**

http://www.softwaretestinggenius.com/articalDetails.php?qry=572#commentsList

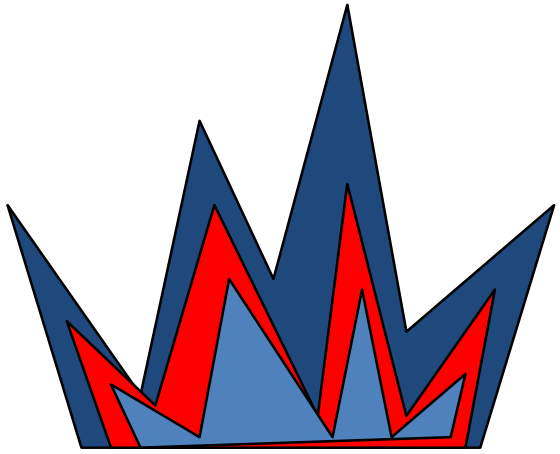
Error - Defect - Failure

an error creator ...

Human



a
defect based on the
error ...



... Lead to failure

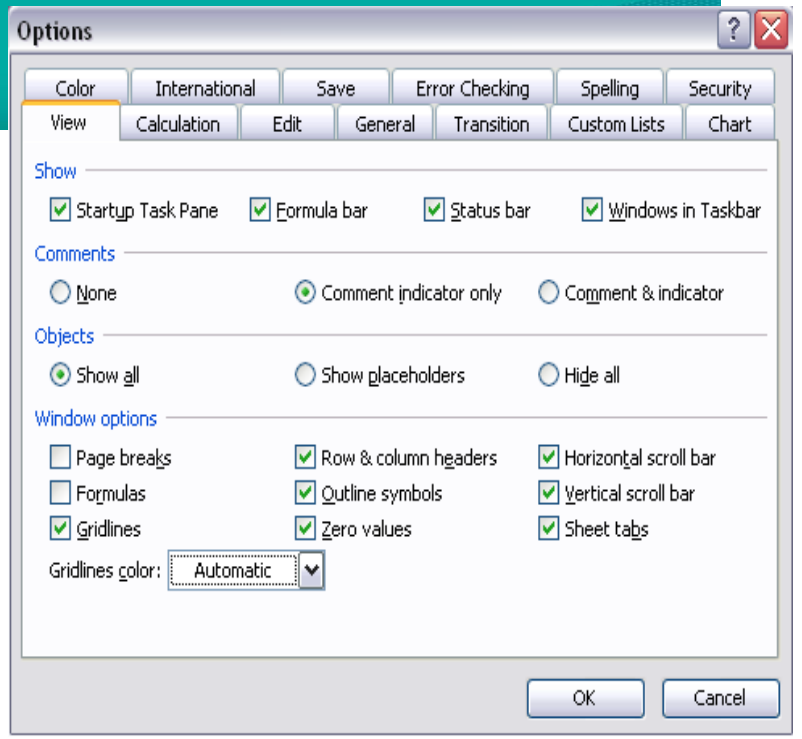
Where are the errors coming from?

- Errors in the Software Requirement Specifications (SRS)
- Software Design Specifications (SDS)
- Development and coding
- Implementation of the software and system
  - Errors in the use of the system
  - Environmental conditions
  - Intentional damage
  - Potential consequences of earlier errors, intentional damage, defects and failures.

Reliability vs faults

- Reliability: The software will not cause the failure of the system for a specified time under specified conditions
  - Can a system be fault-free? (zero faults, right first time) ✗
  - Can a software system be reliable but still have faults? ✓
  - Is a "fault-free" software application always reliable? ✗

Why not just "test everything"?



In order to ensure software reliability, many combinations of possible input parameters, hardware/software environments, and system conditions need to be checked against for conformance.

As example, consider the testing of Microsoft Excel software ...

- Even if only View Tab is considered for testing
- 20 possible configuration to be tested...
- Each configuration can take two possible values (i.e. checked or unchecked)
- Gridlines color can have 56 values..

There are 2^{20}x56 = 58,720,256 combinations to be tested.... resulting into combinatorial explosion of test cases.

If 5 minutes is required to execute one test case, then it would require 559 years to complete all tests merely for View tab option!

Complete and exhaustive testing is impossible.....

Exhaustive testing?

- What is exhaustive testing?
  - when all the testers are exhausted ✗
  - when all the planned tests have been executed ✗
  - exercising all combinations of inputs and preconditions ✓
- How much time will exhaustive testing take?
  - infinite time ✗
  - not much time ✗
  - impractical amount of time ✓

How much testing is enough?

- it's never enough ✗
- when you have done what you planned ✗
- when your customer/user is happy ✗
- when you have proved that the system works correctly ✗
- when you are confident that the system works correctly ✗
- it depends on the risks for your system ✓

http://www.softwaretestinggenius.com/articalDetails.php?qry=572#commentsList