TEST CASE DESIGN

What is Good Testing?

- We cannot test everything
- Good testing relies on a good selection of test cases

How to specify a test case?

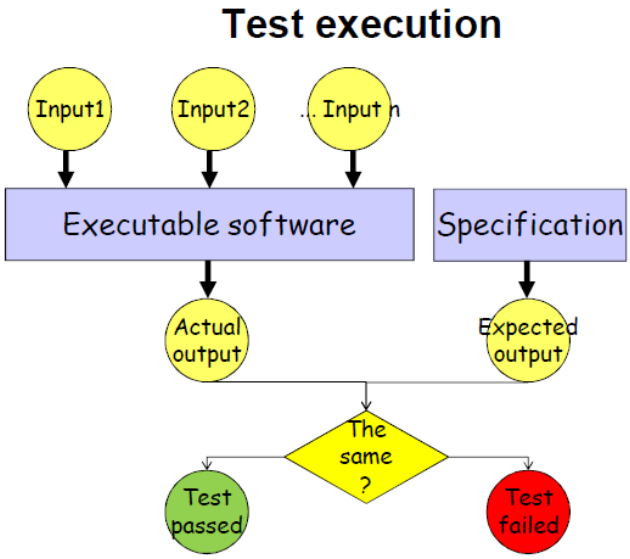
- Testing is the process of executing a program with the intent of finding errors.
- Test failure occurs when fault executes
- How do we detect that failure has occurred?
- The actual output is different than the expected output
- How do we obtain the actual output?
- We execute a program with a given set of inputs

How to specify a test case?

- For the full specification use a Test Case Specification Template (e.g. From IEEE 829 standard)
- Simple (tabulated) specification has to have as a minimum the following entries
  - Test case ID
  - Specific values for all the inputs
  - Expected output(s)
  - Actual output(s)
  - Observations

How to specify a test case?

- For the full specification use a Test Case Specification Template (e.g. From IEEE 829 standard)
- Simple (tabulated) specification has to have as a minimum the following entries
  - Test case ID
  - Specific values for all the inputs
  - Expected output(s)
  - Actual output(s)
  - Observations



The triangle problem (simple version)

- The input to the `TriangleType` function are three numbers a, b and c that represent the lengths of the three sides of the triangle.
- Based on these inputs the function determines the type of the triangle, which can be
  - Equilateral (i.e. all three sides are equal)
  - Isosceles (two equal sides)
  - Scalene (three unequal sides)
- The function returns the result in the form of the character string, e.g. 'Equilateral' if the triangle is equilateral.

Class exercise

Suggest a set of test cases for testing the `TriangleType` function

Answer – Minimum Tests

Test case	Input values	Expected results
1	3, 3, 3	Equilateral
2	3, 3, 2	Isosceles
3	3, 4, 5	Scalene

Answer – MoreTests

Tests	Inputs	Expected Results
Any element of an input set is negative	(-1,1,1), (1,-1,1), (1,1,-1)	Error Message
Any element of an input set is zero	(0,1,1), (1,0,1), (1,1,0), (1,0,0), (0,1,0), (0,0,1), (0,0,0)	Error Message
Any element of an input set is an alphabetic letter	(T,1,1), (1,T,1), (1,5,T)	Error Message
Any element of an input set is a character symbol	(:,1,3), (5,., 8), (2,4,/)	Error Message
There are less than three elements of the input set	(1,2)	Error Message
There are more than three elements of the input set	(2,4,5,8)	Error Message
Any element of an input set is a combination of any of the invalid inputs	(1A, 3,4), (4,/W",8) (2,4,:)	Error Message
The sum of two numbers equals the third in any of three permutations a+b=c, a+c=b, b+c=a	(2,4,6), (4,9,5), (8,4,4)	Error Message
The sum of two numbers is less than the third in any of three permutations a+b<="" a+c	(3,3,8), (2,5,1), (7,3,3)	Error Message

Lessons learned

- Specification must be as precise as possible
- Testing must include negative cases
- Effective testing requires careful selection of appropriate test cases
- Accurate and thorough specification is essential
- It helps to have background knowledge, both of the application area to be tested and of computer programming

SOFTWARE RISKS & RISK BASED TESTING

What is Software Risks?

The possibility that the system or software might fail to satisfy some reasonable customer, user, or stakeholder expectation.

- Unsatisfactory software might omit some key function that the customers specified, the users required or the stakeholders were promised.
- Unsatisfactory software might be unreliable and frequently fail to behave normally.
- Unsatisfactory software might fail in ways that cause financial or other damage to a user or the company that user works for.
- Unsatisfactory software might have problems related to a particular quality characteristic, which might not be functionality, but rather security, reliability, usability, maintainability or performance.

What is Risk Based Testing?

- Risk-based testing is the idea that we can organize our testing efforts in a way that reduces the residual level of product risk when the system ships.
- Risk-based testing uses risk to prioritize and emphasize the appropriate tests during test execution.
- Risk-based testing starts early in the project, identifying risks to system quality and using that knowledge of risk to guide testing planning, specification, preparation and execution.
- Risk-based testing involves both mitigation - testing to provide opportunities to reduce the likelihood of defects.
- Risk-based testing can also involve using risk analysis to identify proactive opportunities to remove or prevent defects.

Case Study

- You have been request to revamp of www.gov.lk portal.
- Discuss the problem
- Identify Software Risks and Mitigation plan for testing.

Email your answers to
kapilalk@gmail.com

Identify Product Risks of GOV.LK

No	Key Product Risks
1	Functionality of following core areas might not work properly. - Services Search - Poll - News Feed
2	Social Media integration might not work properly
3	Translatability might not work properly for all three languages
4	Portal might not work properly in all key browsers
5	Portal might not work properly in mobile devices
6	Application might not able to handle user loads in peak times
7	UI theme might not align with standards

Identify Mitigation plan for Product Risks

No	Key Product Risks	Criticality	Mitigation Plan
1	Functionality of following core areas might not work properly. - Services Search - Poll - News Feed	High	Perform **Functional Testing** for core areas based on optimized test design.
2	Social Media integration might not work properly	Medium	Perform **Integration Testing** for all social media integrations
3	Translatability might not work properly for all three languages	High	Perform **Translatability and Localization Testing** on translatable content
4	Portal might not work properly in all key browsers	High	Perform **Cross Browser Testing** on identified core browsers
5	Portal might not work properly in mobile devices	Medium	Perform **Mobile Compatibility** testing on identified mobile devices
6	Application might not able to handle user loads in peak times	High	Perform **Performance Testing** based on forecasted user load growth.
7	UI theme might not align with standards	Low	Perform **UI Testing** for against agreed standards,

TEST PLANNING

Agenda

- Need of Test Strategy for industry
- How to develop a Simple Test strategy
- Industry Standards for test planning
- Use of IEEE guidelines for test planning
- How to develop a Simple test plan
- Case Study

Test Strategy & planning Case Study

- You have been request to revamp of www.gov.lk portal. You already discussed the problem & identified Software Risks and Mitigation plan for testing.

- Derive Test Strategy based on above risks and their mitigation.

Derive Test Strategy for GOV.LK

- Functional Testing
- Translatability and Localization Testing
- Cross Browser Testing
- Performance Testing
- Mobile Compatibility
- Integration Testing
- UI Testing

IEEE Test Plan Structure

IEEE 829 Test Plan Structure

- Test plan identifier
- Introduction
- Test items
- Features to be tested
- Features not to be tested
- Approach
- Item pass/fail criteria
- Suspension criteria and resumption requirements
- Test deliverables
- Testing tasks
- Environmental needs
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies
- Approvals

Thank You