01/09/14

ISTQB Testing

> Schedule

Fundamentals of Testing

Lifecycle of Testing

ISTQB Foundation:

Learn the fundamental testing principles including details of the testing process

Understand the testing throughout the project development lifecycle

Apply a variety of test techniques of the more effective test preparation

Managing your own testing process.

Fundamentals of Testing

Obejectives

> Understand the need for testi ng, and explain with examples why testing contributes to better qualuit software

> Know the common objectives of testing

> Explain the effects a software defect can cause, distingusing between cause and effect

> Defect, error and fault

> Testing =/= debugging

Recall the seven principles pof tesint and the fice fundamental

Introduction

>Why do you think we need software testing?

Gives something that the client has asked for. Improve the quality of the software.

To help clarify requirments!

>What is quality?

Software that satisfies the client’s needs and works.

Why do we test?

> Testing and Risk?

How are we going to mitigate risk. Are we happy with the level of risk

> Testing and Quality?

> Exhaustive Testing?

Resources and Time is a factor

Question Time

B

Key Testing Term

>Defect

>Effects a defect might have on a project

Defect can lead to failure/quality

> Root cause vs. Effect

Don’t want the root cause to keep effecting\*

>Error

Human Error > Software Defect > System Failure

>Failure

>Fault

>Mistake

>Bug

Look through the guide for the definition of each,

Question Time

C

Debugging and Testing

Debugging – used by developers to identify the cause of bug or defects in code and undertake corrections

Testing – systematic exploration of a component or system with the main aim of finding and reporting defects

The Seven Testing Principles

1. Testing shows the presense of bugs
2. Exhaustive testing is impossible
3. Early Testing
4. Defect Clustering
5. The Pesticide paradox

> Testing no longer becomes relevant with multiple uses of same test, must adapt

1. Testing is context development
2. Absence of errors fallacy

Developer and Tester Mindset

>The approach of a tester

>The approach of a developer

>Using these different viewpoints to mitigate risk and increases

Summary

Go over the key terms, be aware of them.

Some of the reasons as to why we

Part 2 Testing Through the Lifecycle

Objectives

>Explain the rellationshiip with examples, between development, test activities and work products in the SDLC

> compare the different levels of testing activities with reference to major objectives, typical objects and targets of testing and types of defects and failure  
> Examine software development models (V Model, Waterfall and Itrative) and and apply best practice to real IT projects and products

> Adapt the key software development models to the project context and product and characteristics.  
> Remember that characteristics of good testing apply to ALL development models

Fundamental Test Process

> Test Planning and Control

- Determines what is going to be tested

- How it is going to be tested

- How it will be achieved -> is it achieable

- Determining your exit criteria

-> What are we gonna test,

> Test Analysis and Design

- The fine detail of what to test

- Test conditions, cases and procedures

- The Test Basis is reviewed

- We look at how to combine test and conditions into test cases

> Testing Implementation and Execution

- Prioritising test cases

- Creating test suites from collect test cases

- Environment set-up

- Log testing activities and defects

- Running tests

> Evaluating Exit Criteria and Reporting

- Has the criteria been met to satisfy the stakeholders?

- Does the system do as expected?

- Determine if more tests need to be made?

- Is the system in a ready state for release?

> Test Closure Activities

- Make sure docs are up to date and archived

- Passing over testware to maintenance teams if available

- Lessons learnt

Development Models

- A model is a representation of a real event. It allows us to define a high level concept and help others to understand what we want to achieve.

The Methodology you choose depends on factors such as:

>The nature of the project

>The project schedule

>Resource availability

-> theres flexibility across all methodologies if applied correctly

Waterfall

V-Model

Requirement -> Functional -> Technical -> Program -> (Specification) Coding

Acceptance -> System -> Integration -> Unit (Testing)

Pros: Higher chance of success as test plans are developerd earlier

Defects found earlier

Works very well o smaller projects

Cons: rigid in exectuont

No early prototype

Question Time:

Verification and Validation

**Continous Verfication** is to check if the documents are current, no information has been lost, contain any ambuguirites or errors. As these will cause problems if unchecked

**Validation** is to check the documents against the original User Requirements to make sure that requirments are still being satisfied – Does it still do what the user wants

Iterative-Incremental Development Models

Pros: Small and frequent improvements

-fast deployment

-team skills mprovement

Cons: light documentation

-formal records of change not be created

- Regression testing may get out of control

Quiz Time

1. Resources available -> Time/Money/Context of the System being developed
2. Give people a framework to work by, allowing for better understanding

> good visual representation, here to encourrage

1. Planning and Control -> Test Design and Analysis -> Implementation and Execution -> Exit Criteria and Review -> Closing Test Activities
2. Lots of documentation; Definitive ending time; Everything gets completed before the next stage starts
3. That the functions adhere to the specification set for them
4. Little Documentation, Regression Testing out of hand.
5. Allowing The detection of early defects though simultaneous testing and development

> Cleary recognizable testing

> Helps recognize testing as a very main part of it:

1. Cons: Rigid/Resource Development. Pro: Testing can start straight away, Customer involvement

Summary