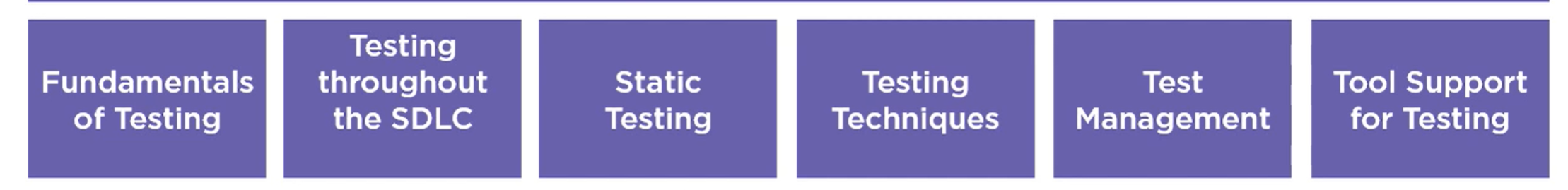
It doesn’t matter how good you are at automation if you don’t write the tests right

Structured quality tests to find bugs

6 topic:



Sub topic – what is testing and why is it necessary

One course for each chapter on pluralsight

What is Testing?

Why necessary?

Howe does it contribute to the end of product?

The seven testing principals!

The test process – organisation

The psychology of testing

Graphical user interface, text, application

Description automatically generated

Checking results

* Test planning
* - what how by when
* Do these requirements make sense
* How to design good or bad tests
* How to execute them
* Reporting bugs well
* Test reporting

Objectives testing

* Prevent defects by evaluating requirements
  + You want to get the bug BEFORE it happens, evaluate the requirements before the code is written and identify contradictions
  + Ask are all requirements logically sound?
  + Reacting not as good as preventative
* To find defects and failures and so reduce the risk of inadequate software
  + Somew flaws can be tolerated some can not
* Have all requirements been met?
* Check whether the tesxt object is complete and validated so it works as users/stakeholders expect
* To build confidence in ther quality of the object
* To provide sufficient information for stakeholders to make informed decisions about their software
  + Evey piece of software has bugs- you need to know what they are
  + Perfect is the enemy of the good- you need to be able to say to management what the bugs are so they can deliver an informed decision on whether to deliver and how to budget
* To comply with contractual, legal or regulatory standards
  + It is your job to ensure your software follows laws and standards and to prove that with testing

What testing is not:

* Testing is not debugging!
  + Tests shows failures caused by defects in the software – debugging is carried out by developers to find and fix the underlying cause
  + Basically testing is finding issues – debugging is finding the cause of the issue
  + In testin you say: Whsat you did, what happened, what should have happened

7 Major testing principals

* Test early, test often

**Need to be able to answer:**

* What is testing?
  + Includes many activities – a way of proving that things work, building trust in a product, communicating to managers the stage of a product development and ensure ing it meets standards given and that those standards are sensical
* What activities does testing involve?
  + Test planning: What, How, By When?
  + Analysis – Do these requirements make sense?
  + Test design – How to tell between a good or bad test
  + Executing tests
  + Bug reporting – what happened and under what circumstances
  + Test reporting – what does the test evidence that things work look like, what bugs are there and how serious?
* What are the typical testing objectives?
  + Preventing defects
  + Finding software defects
  + Verifying the requirements are met
  + Does the software work as expected?
  + Building confidence
  + Provide information – to allow for informed decisions
  + To ensure legal/contractual requirements are provably met
* What testing is NOT
  + Debugging

Module 2.

* Why testing is necessary?
  + Malfuction can cause costly/dangerous error
* Error vs. defect vs. failure
* Root cause of a fauilure vs it’s effects

Points for testing:

* Requirement review
* Works with system designers – make room for testing for each element
* A picture containing text, businesscard, screenshot

  Description automatically generatedA picture containing text, screenshot, businesscard

  Description automatically generated
* This way you can create backdoors for testing in the code
* Work closely with developers while they code
* Testers verify and validate code before release

**Know the difference for the exam!**

Quality assurance

Quality control

Testing

Quality management

* Quality assurance and quality control
* Quality assurance
  + Proper processed- are we following protocol?
  + When things go wrong – process to find root cause
  + Managing tool – proactice preventing defects
* Quality Control
  + Testing, test design, execution…
  + Find defects that you can’t prevent, reactive
  + Propertesting
  + QC is part of qa
  + Graphical user interface, application

    Description automatically generated

Error

* Mistake – human error, misunderstanding requirements for example
* Errors can lead to faults or defects

Defect

* Alswo known as a fault or bug
* Bugs can trigger further bugs
* Human mistake built into the software
* Bugs **may** lead to failure

Failure

* When you use the software and it doesn’t do what you want it to do
* **Ugs may only be manifest** after a specific sequence of events or accessed with a specific sequence of actions

Graphical user interface

Description automatically generated

REMEMBER dif for exam!

Why do mistakes happen?

* Time pressure
* Human fallibility
* In experience or lack of skill
* Complexity – more complex more mistakes
* Misunderstandings – never assume
* New or unfamiliar tech – someone has to trial and error

**Treat the cause, not the symptoms**