

# ARCHITECTURE AND TESTING

Andre Kurnia - 00000021269  
Richard David - 01082180003



# WHAT AND WHEN?

- Testing can be seen as the process of making sure that a software system meets its requirements
- Making sure the software brings the necessary functionality to its user community
- Architecture can help make testing less costly and more effective when embraced in testing activities
- Architecture defines the units that are to be tested
- Architecture defines the responsibility and interactions of the units
- Testing and its relations to the architecture are defined in each levels of testing:
  - Unit testing; when a specific piece of software is ready to be tested
  - Integration testing; when separate software units are combined
  - Acceptance testing; when deployed to a specific set of end users
  - Regression testing; when a change has been made to the system



# **LEVELS OF TESTING .XX**



# UNIT TESTING

- Runs on specific pieces of software
- Typically written by developers
- A precondition for that unit to be delivered to integration activities
- Provide confidence that each system's building blocks exhibit correctness
- A unit corresponds to an architectural element in one of the architecture's module views
- Architect roles in unit testing:
  - First, it defines the units: they are architectural elements in one or more of the module views.
  - Second, it defines the responsibilities and requirements assigned to each unit.

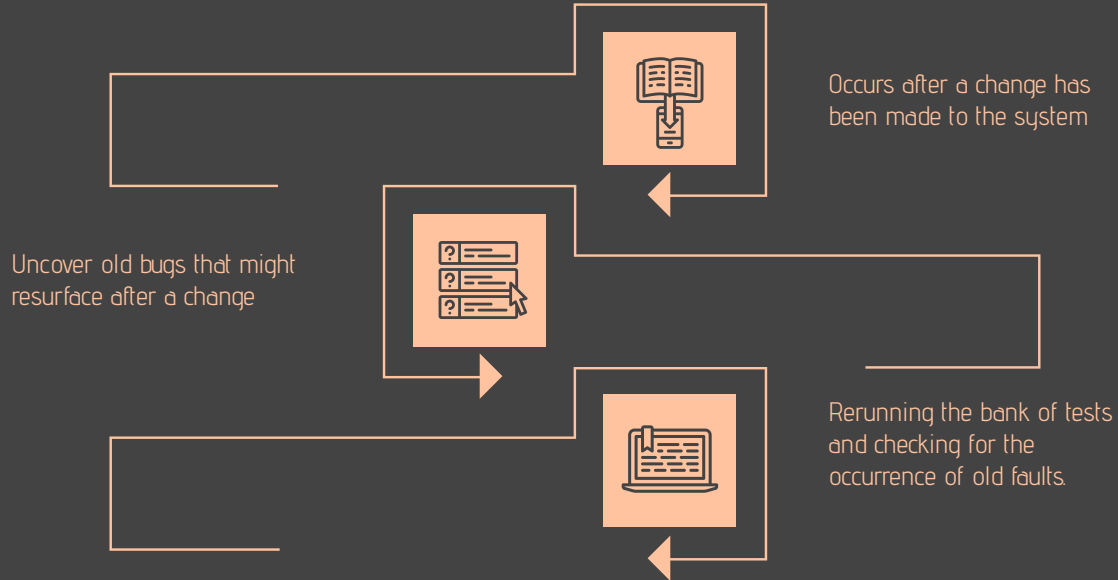
# INTEGRATION TESTING

- What happens when separate software units start to work together
- Concentrates on finding problems related to the interfaces between elements in a design.
- At the end of integration testing, the project has confidence that the pieces of software work together correctly and provide at least some correct system-wide functionality
- Architect roles in integration testing:
  - First, the increments that will be subject to integration testing must be planned, and this plan will be based on the architecture
  - Second, the interfaces between elements are part of the architecture, and those interfaces determine the integration tests that are created and run.

# ACCEPTANCE TESTING

- Performed by users
- Often in the setting in which the system will run
- Two special cases: Alpha Testing and Beta Testing
  - Alpha testing usually occurs in-house
  - Beta testing makes the system available to a select set of end users
- Architect roles in acceptance testing:
  - Stressing the system's quality attribute behavior

# REGRESSION TESTING



The background is a solid light orange color. It features stylized, darker orange hills and foliage. At the bottom, there are two large, rounded hills. On the left and right sides of these hills, there are dark orange, leafy branches. Above the main text, there are smaller, more distant hills. The text is centered and reads: 

# **BLACK-BOX, WHITE-BOX AND RISK-BASED TESTING**



# BLACK-BOX TESTING

- Treats the software as an opaque “black box”
- Not using any knowledge about the internal design, structure, or implementation
- Only source of information about the software is its requirements
- Architect roles in black-box testing:
  - Creating architecture document where the requirements for a piece of the system are described

# WHITE-BOX TESTING

- Makes full use of the internal structures, algorithms, and control and data flows of a unit of software
- Exercise all control paths of a unit of software

# GRAY-BOX TESTING

- Lies between black-box and white-box testing
- Testers get to avail themselves of some, but not all, of the internal structure of a system

# ADVANTAGES AND DISADVANTAGES

- Black-box testing
  - Advantages: Not biased by a design or implementation, and it concentrates on making sure that requirements are met
  - Disadvantages: Inefficient
- White-box testing
  - Advantages: Keys in on critical errors more quickly
  - Disadvantages: May suffer from a loss of perspective


# RISK-BASED TESTING

- Concentrates effort on areas where risk is perceived to be the highest
- Factors include: immature technologies, requirements uncertainty, developer experience gaps
- Architects can identify areas where architectural decisions would have a widespread impact
- If architecturally significant requirements are not met, then the system is unacceptable



# TEST XX. ACTIVITIES

Fun fact : testing can consume from 30 to 90 percent of developments schedule and budget

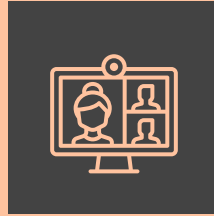


# THERE ARE MANY THINGS YOU COULD DO IN TESTING!



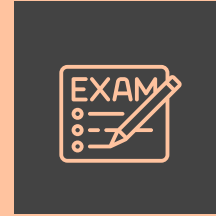
## TEST PLANNING

Well planned and appropriate resources become the main aspect because it includes time schedule, labor to run and technology\* that the architecture use.



## TEST DEVELOPMENT

Test procedures with scripted cases that chosen will be implemented in this state.



## TEST EXECUTION

Call all the testers! Because it's the testers time to apply the tests to the software and capture + record errors that shown up!

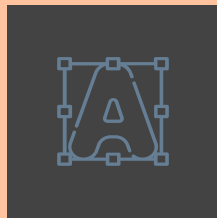
\*  
Technology might include test tools, automatic regression testers, test script builders, test beds, test equipment or hardware such as network sniffers, and so forth.

# THERE'S SEVERAL THINGS YOU SHOULD DO IN TESTING



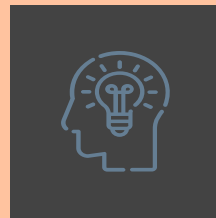
## TEST REPORTING

After testing the the software, it's the time for them to report overall metrics about the test results to the project's technical management.



## DEFECT ANALYSIS

Analyse the report, and make a decision about whether the software is ready for release or not!



## TEST HARNESS CREATION

Setting up the environment for the elements to be tested along with controlling their state and the data flowing into and out of the elements



# ARCHITECT IS EVERYTHING!

Architect plays a role and inform every each activities! It's really important to have architecture in this state. Here's what architecture able to do..

- Contributing to give useful information and suggestions back-to-back
- Providing the list of software units and incremental subsets
- Providing insights as to the complexity or future plannings
- Suggesting useful test technologies that will be compatible with the architecture



# THE ARCHITECT'S ROLE





# 01

## **DESIGN THE SYSTEM**

Work with the test team to establish what is needed.

# 02

## **DEFINED THE REQUIREMENTS**

By using scenarios through many testing, the team can defined the testability reqs clearly.

# 03

## **GIVE ACCESS TO TESTERS**

Insure that testers have access to the documents that might helps you to earn some feedbacks.

# 04

## **GIVE TESTERS OPTIONS**

By giving many options either different versions of the app,, we gain benefit to developers to gain comparing results from the testers directly.



## — LAST BUT NOT LEAST TO REMEMBER

As a practical matter, the architect should be the one who will diagnose the problem when something goes seriously wrong. That's why it requires many people to be involved for analysing and comparing the next update.

# ARCHITECTURE PLAYS A KEY ROLE IN BOTH IMPLEMENTATION & TESTING

How architecture determines what is being tested, developing quality attributes to be tested means architecture become the main aspect to be the most versatile one. It appears architecture can be benefit for cost-related and schedule for development



**THANK YOU**

