Integration, System and Other Types of Testing

Davide Falessi

Integration Testing

- Once you have tested the single units, these have to be integrated which gives raise to integration testing.
- The purpose of integration testing is to verify the functional, performance, and reliability **between the integrated units**.

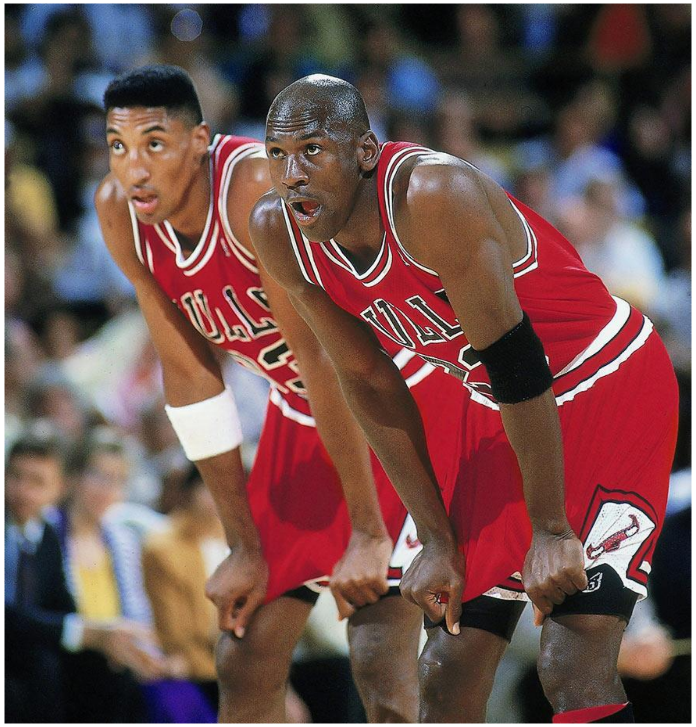
Integration testing gone wrong 1

Integration testing gone wrong 2

Integration testing gone wrong 3

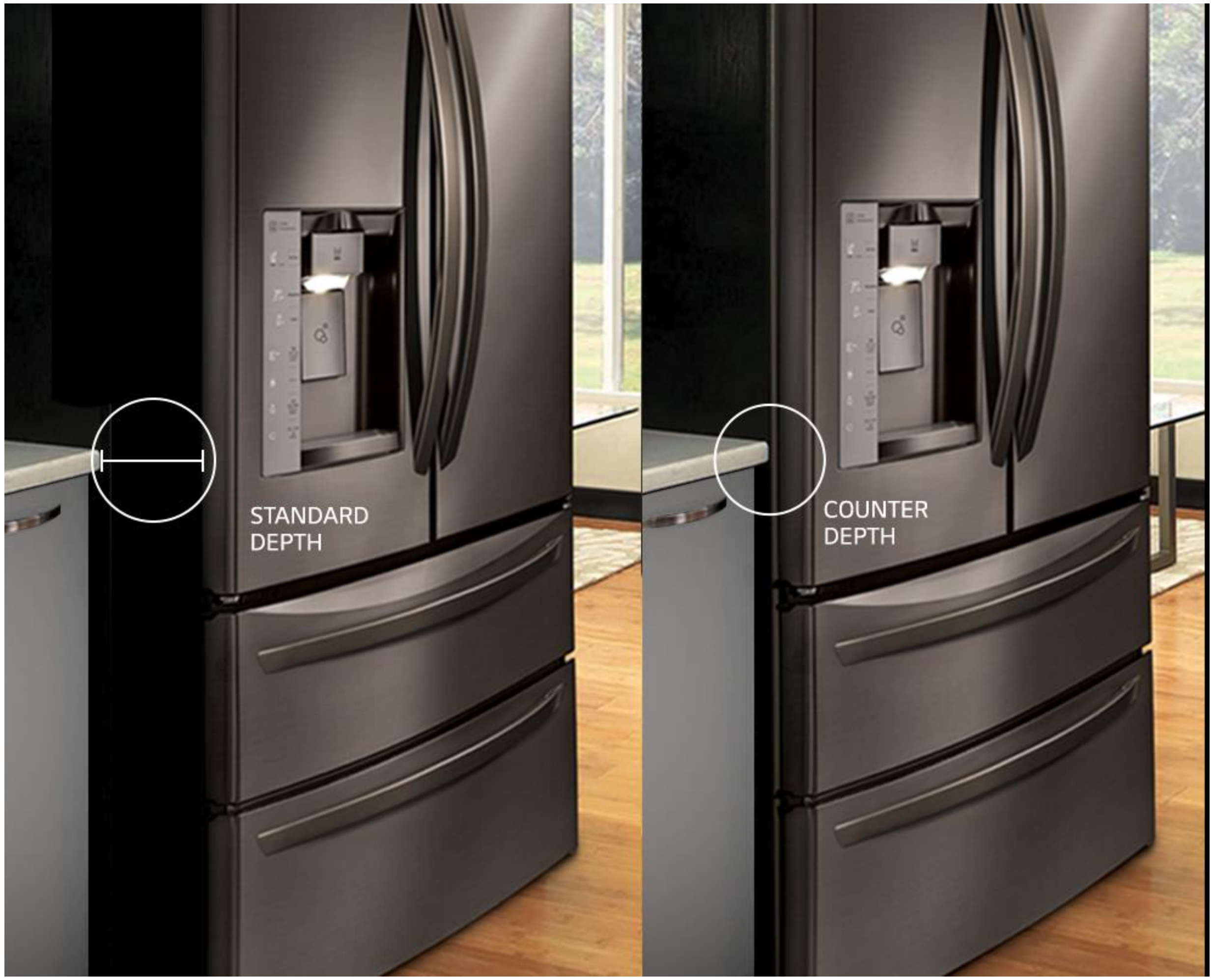
- https://twitter.com/yogthos/status/951905438727057408

Integration gone great!



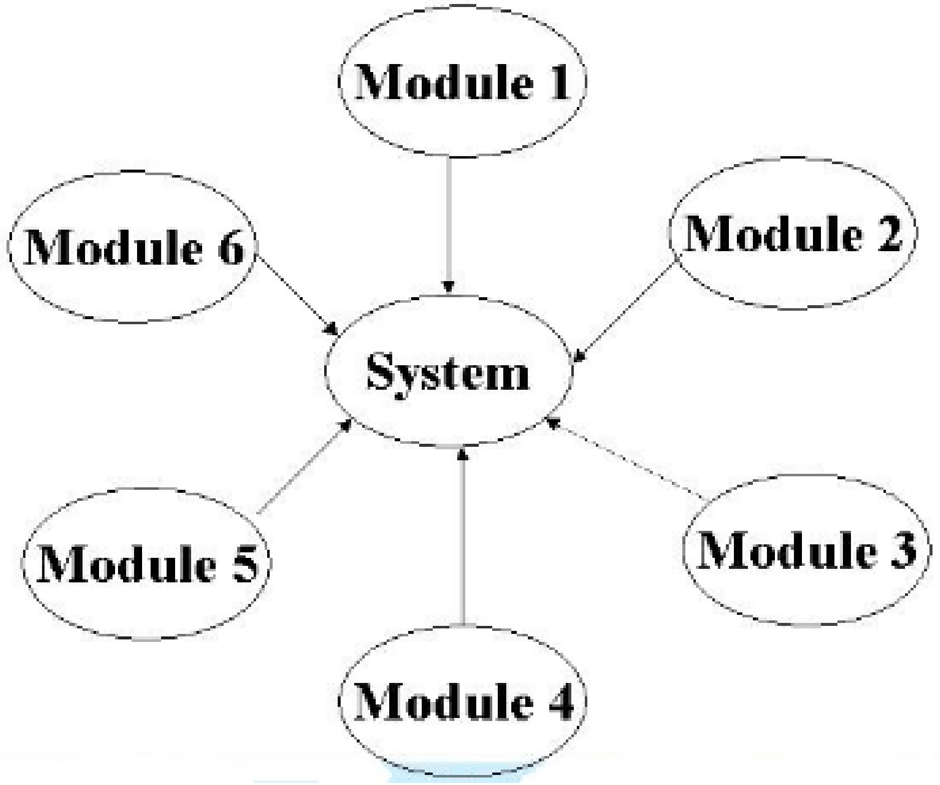
Integration testing gone great!

Integrability commercial



IT: Big bang

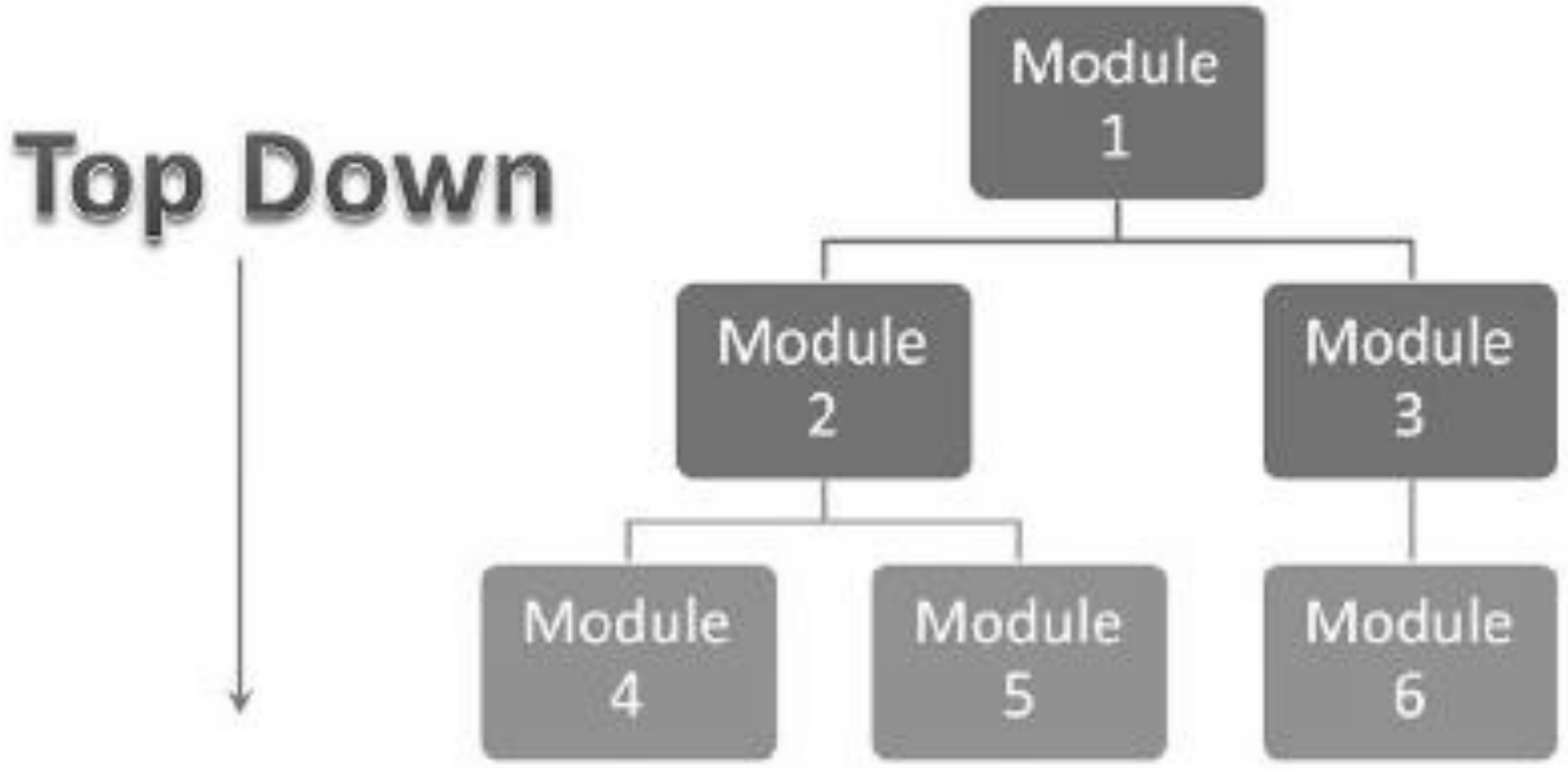
- In Big Bang integration testing all components or modules are integrated simultaneously, after which *everything is tested as a whole*.



IT: Big bang

- **Pros:** everything is finished before integration testing starts.
- **Cons:** in general it is time consuming and difficult to trace the cause of failures because of this late integration.

IT: Top down



IT: TOP DOWN

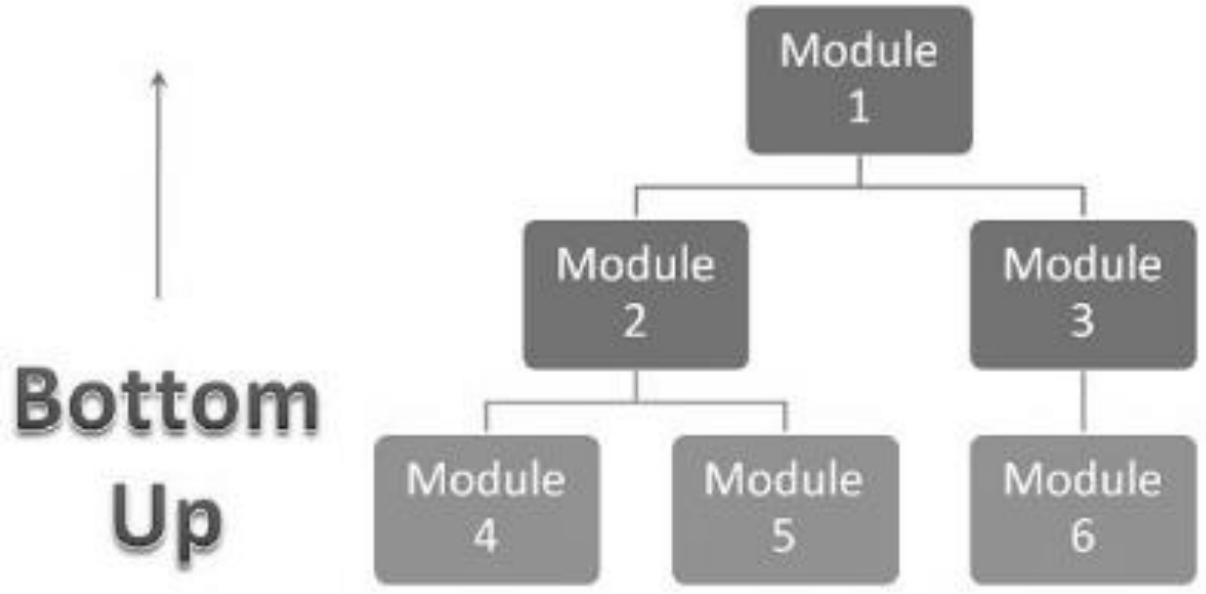
- Testing takes place from top to bottom, following the control flow or architectural structure (e.g. starting from the GUI or main menu).
- Not developed components or systems are substituted by stubs*.

- **Stubs**: a piece of code used to simulate (in an easy/**dummy** way) the behavior of existing code (such as a procedure on a remote machine) or be a temporary substitute for **yet-to-be-developed code**. The code under testing calls the stubs.

IT: Top down

- **Pros:**
  - The tested product is very consistent because the integration testing is basically performed in an environment that almost similar to that of reality
  - Developing stubs is cheaper than drivers.
- **Cons:**
  - Basic functionality is tested at the **end** of cycle

IT: Bottom up


Bottom Up
Module 1
Module 2
Module 3
Module 4
Module 5
Module 6

IT: Bottom up

- Testing takes place from the bottom of the control flow upwards. Not developed components or systems are substituted by drivers*.
- **Driver**: **dummy programs which are used to call the functions** of the lowest module in case when the calling function does not exists.

IT: Bottom up

- **Pros:**
  - In this approach development and testing can be done **together** so that the product or application will be efficient and as per the customer specifications.
- **Cons:**
  - Developing **drivers is more expensive than stubs**.
  - We can catch the key interface defects at the end of cycle.
  - It is required to create the test drivers for modules at all levels except the top control.

IT: INCREMENTAL

- All units are integrated **one by one**, and a test is carried out after each step.
- **Pros**:
  - defects are found **early** in a smaller assembly when it is relatively easy to detect the cause.
- **Cons**:
  - It can be **time-consuming** since stubs and drivers have to be developed and used in the test.

System Testing

- The process of testing of an integrated hardware and software system to verify that the system meets its specified **requirements**.
- It evaluates working of system from **user** point of view, with the help of specification document.
- It does not require any internal knowledge of system like design or structure of code.

ST: Advantages

- It is very important to complete a full test cycle and ST is the stage where it is done.
- It is performed in environment which is similar to the production environment and hence stakeholders can get a good idea of the user’s reaction.
- It helps to **minimize after-deployment troubleshooting** and support calls.

ST: How To

- Requirements and expectations should be clear and the tester needs to understand real time usage of application too.
- Examples:
  1. If the site launches properly with all the relevant pages, features and logo
  2. If the user can register/login to the site
  3. If the user can see products available, can add products to his cart can do payment and can get confirmation via e-mail or SMS or call.
  4. If the major functionality like searching, filtering, sorting, adding, changing, wish list etc work as expected
  5. If number of users (defined as in requirement document) can access the site simultaneously
  6. If the site launches properly in all major browsers and their latest versions

INTERFACE TESTING

- It is performed to evaluate if units pass data and control correctly to one another.
- It aims at verifying that all the interactions between these modules are working properly and errors are handled properly.

(to be continued…)

Internationalization Testing

- It verifies that the application under test works across multiple regions and cultures. It checks if the code can handle all international support without breaking functionality that might cause data loss or data integrity issues.

Inter Systems Testing

- An application can be hosted across different locations; however, all data needs to be deployed over a central location.
- Inter system testing aims at ensuring correct data flow across locations.

Load Testing

- Load testing is performance testing technique using which the response of the system is measured under various load conditions. The load testing is performed for **normal** and **peak** load conditions.
  - Evaluate performance acceptance criteria
  - Identify critical scenarios
  - Design workload Model
  - Identify the target load levels
  - Design the tests
  - Execute Tests
  - Analyze the Results

Loop Testing

- Guidelines: try to design a test in which:
  - the loop body isn't executed at all.
  - the loop body is executed exactly once.
  - the loop body is executed exactly twice.
  - the loop body is executed some ``typical'' number of times.
  - If there is an upper bound, n, on the number of times the loop body can be executed, then the following cases should also be applied:
    - the loop body is executed exactly n-1 times.
    - the loop body is executed exactly n times.
    - the loop body to be executed exactly n+1 times.

Negative Testing

- It ensures that the application under test does NOT fail when an unexpected input is given.
- Examples:
  - Embed Single Quote on URL when it tries to query the database.
  - Skip the Required Data Entry and try to proceed.
  - Verify each Field Type Test.
  - Enter large values to test the size of the fields.
  - Verify the numeric boundary and numeric size test.
  - Verify the Date Format and its validity.
  - Verify the web session and check for the performances.

Reliability Testing

- It tests a software's ability to function given environmental conditions consistently.
- Dependent elements of reliability Testing:
  - Probability of failure-free operation
  - Length of time of failure-free operation
  - The environment in which it is executed