INTEGRATION TESTING

PBA SOFTWAREUDVIKLING/
BSC SOFTWARE DEVELOPMENT

Christian Nielsen cnls@cphbusiness.dk

Tine Marbjerg tm@cphbusiness.dk

SPRING 2019

TODAY’S TOPICS

- **Overview**
  - Learning objectives
  - Introduction
  - Production database / Test database
  - MySQL
  - DBUnit
  - H2
  - Mockito
  - Examples
  - Exercises
  - Assignment

LEARNING OBJECTIVES

- Understand integration testing in general
- Know different types of and approaches to integration testing
- Handle production and test databases separately
- Automate integration tests using test databases, database testing frameworks, in-memory databases and mocking

INTEGRATION TESTING

Integration testing is testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.

Phase in software testing in which individual software modules are combined and tested as a group that occurs after unit testing and before system testing.

Assemblages / Groups of units

Test whether all the components within assemblages interact correctly.

If all units work individually, why doubt that they work together?

The point where objects interact is major contributor of bugs!

INTEGRATION TESTING

Focus: Interaction and communication between components Integration, not the functionality of either of them

Problem: The greater the scope of integration testing, the harder it becomes to isolate failures to a specific interface

Example: Code that accesses other objects, services, files, databases or other external resources

Implications: Instantiate objects / Mock objects / Known state

INTEGRATION TESTING

Approaches

Integration testing of a software product can be done in a number of ways throughout the development cycle

Different approaches to integration testing:

- Big Bang

  Approach where all or most of the units are combined together and tested at the same time

- Top-Down

  Approach where top level units are tested first and lower level units are tested step by step after that

- Bottom-Up

  Approach where bottom level units are tested first and upper level units step by step after that

- Sandwich / Hybrid / Mixed

  Approach which is a combination of Top-Down and Bottom-Up approaches with above and below a middle layer

INTEGRATION TESTING

Database layer testing

Persistence layer is difficult to test

Persistence layer requires interaction with an external database

Code that accesses the database can be cumbersome

Database access is relatively slow which gives slow tests

A good test is self sufficient and creates all data it needs

Deleting and inserting data for every test may seem like a big time overhead, but it works

Make a setup method that all database tests call to put the database in a known state

INTEGRATION TESTING

Database layer testing considerations

- Production database
- Test database
- SQL Script
- Connector
- Data testing framework
- In memory database
- Mocking

EXERCISES

- **DATAMAPPER**
  - Test database
  - MySQL
  - DBUnit
  - H2
  - Mockito

RESOURCES…

Integration testing

https://www.guru99.com/integration-testing.html

http://tryqa.com/what-is-integration-testing/

https://www.softwaretestinghelp.com/tools/40-best-database-testing-tools/

DBUnit

http://dbunit.sourceforge.net/index.html

http://dbunit.sourceforge.net/intro.html

H2

http://www.h2database.com/html/main.html