# CSE291 - Introduction To Software Engineering (Fall 2018)

Lecture 14

**Software Testing**

# Software Testing

*The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component.*

# Software Testing

Testing can be defined as

*"A process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item"*

According to IEEE standard

# Who Does Testing?

In most cases, following professionals are involved in testing of a system within their respective capacities:

- Software Tester
- Software Developer
- Project Leader/Manager
- End User

# When to Start Testing?

- An early start to testing reduces the cost, time to rework and error free software that is delivered to the client.
- In SDLC, testing can be started from the Requirements Gathering phase and lasts till the deployment of the software.
- It also depends on the development model that is being used.

# When to Stop Testing?

Following are the aspects which should be considered to stop the testing:

- Testing Deadlines.
- Completion of test case execution.
- Completion of Functional and code coverage to a certain point.
- Bug rate falls below a certain level and no high priority bugs are identified.
- Management decision.

# Testing and Debugging

**Testing:**

- It involves the identification of bug/error/defect in the software without correcting it.
- Normally professionals with a Quality Assurance background are involved in the identification of bugs.

# Testing and Debugging

**Debugging:**

- It involves identifying, isolating and fixing the problems/bug.
- Developers who code the software conduct debugging upon encountering an error in the code.
- Debugging is the part of White box or Unit Testing. Debugging can be performed in the development.

# Types of Testing

**Manual testing**

- This type includes the testing of the Software manually i.e. without using any automated tool or any script.
- There are different stages for manual testing like unit testing, Integration testing, System testing and User Acceptance testing.
- Testers use test plan, test cases or test scenarios to test the Software to ensure the completeness of testing.

# Automation Testing

- Automation testing which is also known as *Test Automation*, is when the tester uses another software to test the software.
- This process involves automation of a manual process.
- Integration of computerized tools into the process of software development
- Code auditing
- Coverage monitoring
- Load tests…etc

# Classification According to Testing Concept

- Black Box Testing
- White Box Testing

# Black Box Testing

- The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.
- The tester does not have access to the source code.
- A tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

# Black Box Testing

**Advantages**

- Well suited and efficient for large code segments.
- Code Access not required.
- Clearly separates user's perspective from the developer's perspective.
- Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language or operating systems.

# Black Box Testing

**Disadvantages**

- Inefficient testing, due to the fact that the tester only has limited knowledge about an application.
- Blind Coverage, since the tester cannot target specific code segments or error prone areas.
- The test cases are difficult to design.

# White Box Testing

- White box testing is the detailed investigation of internal logic and structure of the code.
- White box testing is also called glass testing or open box testing.
- In order to perform white box testing on an application, the tester needs to have knowledge of the internal working of the code.
- The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

# White Box Testing

**Advantages**

- As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively.
- Extra lines of code can be removed which can bring in hidden defects.
- Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing.

# White Box Testing

**Disadvantages**

- Due to the fact that a skilled tester is needed to perform white box testing, the costs are increased.
- Sometimes it is impossible to look into every nook and corner to find out hidden errors.
- It is difficult to maintain white box testing as the use of specialized tools like code analyzers and debugging tools are required.

# Black Box vs White Box

| **Black Box Testing** | **White Box Testing** |
|---|---|
| The Internal Workings of an application are not required to be known | Tester has full knowledge of the Internal workings of the application |
| Also known as closed box testing, data driven testing and functional testing | Also known as clear box testing, structural testing or code based testing |
| Performed by end users and also by testers and developers | Normally done by testers and developers |
| Testing is based on external expectations - Internal behavior of the application is unknown | Internal workings are fully known and the tester can design test data accordingly |
| This is the least time consuming and exhaustive | The most exhaustive and time consuming type of testing |
| Not suited to algorithm testing | Suited for algorithm testing |

# Testing strategies

- There are basically two testing strategies:

**Big bang testing**: tests the software as a whole, once the completed package is available.

**Incremental testing**:

- Tests the software piecemeal – software modules are tested as they are completed (unit tests)
- followed by groups of modules composed of tested modules integrated with newly completed modules (integration tests).
- Once the entire package is completed, it is tested as a whole (system test).

# Unit Testing

- Unit testing is performed by the respective developers on the individual units of source code assigned areas.

- The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

# Integration Testing /System testing

- The testing of combined parts of an application to determine if they function correctly together is Integration testing.

**System Testing**

- This is the next level in the testing and tests the system as a whole.
- Once all the components are integrated, the application as a whole is tested to see that it meets Quality Standards.
- This type of testing is performed by a specialized testing team.

# Testing Strategies

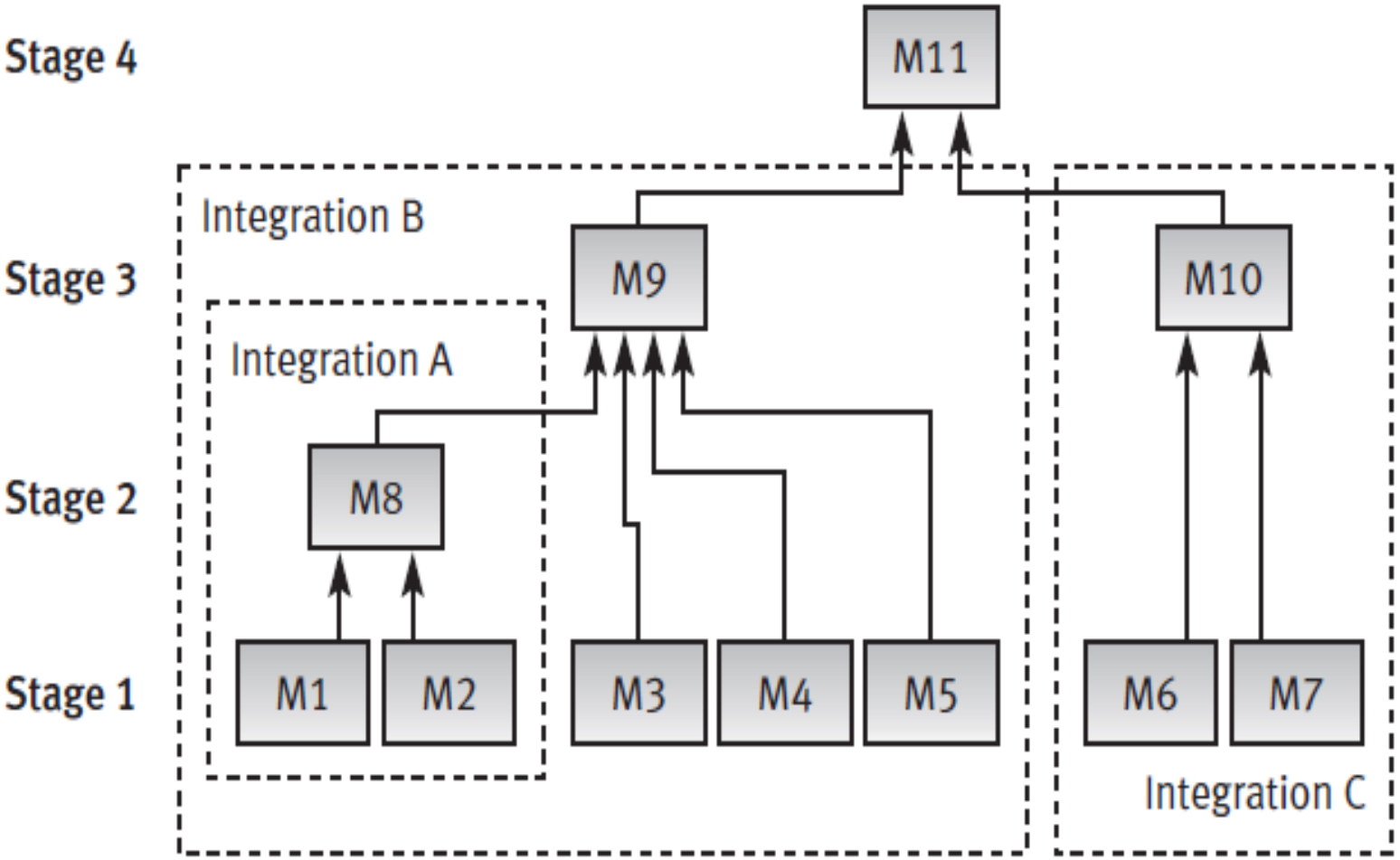
Incremental testing is also performed according to two basic strategies:

- bottom-up and top-down

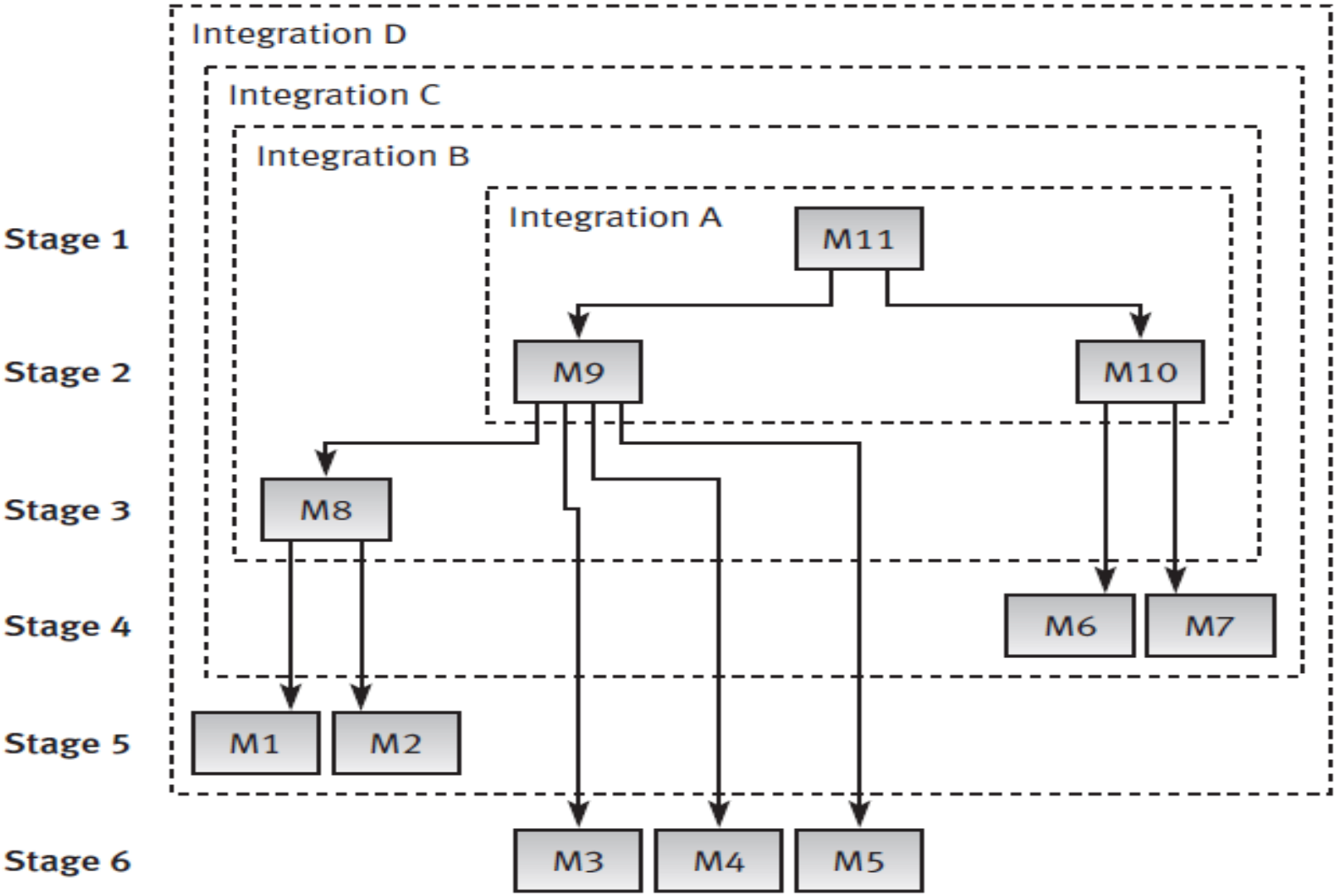
- In **top-down testing**, the first module tested is the main module, the highest level module in the software structure; the last modules to be tested are the lowest level modules.

- In **bottom-up testing**, the order of testing is reversed: the lowest level modules are tested first, with the main module tested last.

# Bottom-up Testing



# Top-down Testing



# Stubs And Drivers For Incremental Testing

- Stubs and drivers are software **replacement simulators** required for modules not available when performing a unit or an integration test.

- A **stub** (often termed a “dummy module”) replaces an unavailable **lower level module**.
- Stubs are used in **top down testing** approach
- when you have the major module ready to test, but the sub modules are still not ready yet.
- So in a simple language stubs are "called" programs, which are called in to test the major module's functionality.

# Stubs And Drivers For Incremental Testing

- A **driver** is also a substitute module but of the **upper level module**
- Drivers are required in **bottom-up testing** until the upper level modules are developed (coded).
- They are used when the sub modules are ready but the main module is still not ready.

# Example of Stubs and Drivers

For Example

we have 3 modules login, home, and user module. Login module is ready and need to test it, but we call functions from home and user (which is not ready). To test at a selective module we write a short dummy piece of a code which simulates home and user, which will return values for Login, this piece of dummy code is always called **Stubs** and it is used in a **top down integration**.

# Example of Stubs and Drivers

- Considering the same Example above:

  If we have Home and User modules get ready and Login module is not ready, and we need to test Home and User modules Which return values from Login module, So to extract the values from Login module We write a Short Piece of Dummy code for login which returns value for home and user, So these pieces of code is always called **Drivers** and it is used in **Bottom Up Integration**

# Regression Testing

- Whenever a change in a software application is made it is quite possible that other areas within the application have been affected by this change.

- The intent of Regression testing is to ensure that a change, such as a bug fix did not result in another fault being uncovered in the application.

# Acceptance Testing

- The most importance type of testing as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirements.

- The QA team will have a set of pre written scenarios and Test Cases that will be used to test the application.

- Acceptance tests are not only intended to point out simple spelling mistakes, Interface gaps, but also to point out any bugs in the application that will result in system crash or major errors in the application.

# Alpha Testing

- This test is the first stage of testing and will be performed amongst the teams (developer and QA teams).
- Unit testing, integration testing and system testing when combined are known as alpha testing.

# Beta Testing

- This test is performed after Alpha testing has been successfully performed.
- In beta testing a sample of the intended audience tests the application.
- Beta testing is also known as pre-release testing.

In this phase the audience will be testing the following:

- Users will install, run the application and send their feedback to the project team.
- Getting the feedback, the project team can fix the problems before releasing the software to the actual users.