



Debre Markos University
Institute of Technology

School of Computing, Software Eng A/program

Software Quality Assurance and Testing (SEng4112)

Chapter Three

Levels of Software Testing

Objective

- **After successful completion of this chapter, students will be able to explain**
 - ✓ **Software testing levels**

Software Testing Levels

- In order to develop products of superior **quality and functionality**, testers and developers use unique and different levels of software testing.
- These levels of testing include **several testing methods** that are used while conducting software testing to gain intended results and to fulfil the demands of the stakeholders.
- Each of these levels of testing has a particular purpose that provides **extra value** to the software development life cycle.

- Levels of software testing are **process** in which every component or small unit of software is **tested**.
- There are different **levels** of software testing; each has its **features**, **advantages**, and **disadvantages**.
- These different levels of software testing are designed to test the software **performance** and **behaviors** of the software at **different stages**.
- There are different stages in the software development lifecycle like **requirements**, **design**, **coding**, and **execution**. **Levels of software testing** are used to find those missing areas between these stages of the **software development lifecycle**.

- There are **four** different levels of software testing, which are as follows:

1. Unit testing
2. Integration testing
3. System testing
4. Acceptance testing

1. Unit Testing

- **Unit testing** is also known as the **first level** of testing.
- A **unit** means a small component.
- **Unit testing** is the process of testing **small components** of software or product.
- This testing is performed using the **white box** testing method.
- One of the main advantages of unit testing is it **allows each small component** of the software to be tested separately.
- The **developers** perform this testing.
- This testing **aims** to test each component of the software to check that
 - it is functioning well or not,
 - does it works as it is intended,
 - does it fulfills the specified requirements.

Advantages of Unit Testing

- **Unit testing** improves the **quality of the code**.
- As **unit testing** **tests only components** of the software, it is **easy** to **debug** the software. If some **error** occurs, only the **latest** changes that are made need to be tested.
- In **unit testing**, the development process is **faster** because it requires fewer efforts to **find** and **fix** bugs than another type of testing.
- Bugs **detected at earlier stage** are **easier to be fixed**.
- Unit testing actually **saves tester's time, effort and cost**. The **cost** of solving defects is **less** as compared with others.
- Unit testing **gives confidence** to the developing team.
- Modules whose unit testing has been done successfully can be **reused** by the developer.

Limitations of Unit Testing

- Unit testing **can't catch every error** in a program. Unit testing by its nature focuses on a unit of code. Hence, it **can't catch integration errors** or broad system level errors.
- It is **not possible to evaluate all execution paths** even in the most trivial programs

Integration Testing

- This is the **second level** of software testing, where each component or unit is combined and tested as a whole.
- The goal behind this testing is to **ensure that software or product is working properly** when the **unit** of software is **combined** together with other components.
- As its name suggests, it is used to identify errors between **interacted components**.
- A typical software project consists of **multiple software modules, coded by different programmers**.
- Integration testing focuses on **checking data communication among these modules**.

Why do Integration Testing?

- ❖ Although each software module is unit tested, defects still exist for various reasons like
 - A module in general is designed by an **individual software developer** whose understanding and programming logic may differ from other programmers.
 - Integration testing becomes necessary **to verify the software modules work in unity**
 - At the time of module development, there are **wide chances of change in requirements by the clients**. These new requirements may not be unit tested and hence system integration testing becomes necessary.
 - Interfaces of the software modules with the **database could be erroneous**
 - External **hardware interfaces**, if any, could be erroneous
 - **Inadequate** exception handling could cause issues.

Approaches of Integration Testing

- The software industry uses variety of strategies to execute integration testing:
 1. Big Bang Approach
 2. Incremental Approach- which is further divided into
 - ✓ Top Down Approach,
 - ✓ Bottom Up Approach and
 - ✓ Sandwich Approach - Combination of top down and bottom up approaches

Big Bang Approach

- Here all components are integrated together at once, and then tested.
- In this testing, the individual modules are not integrated until all the modules are ready.
- In this type of testing, some disadvantages might occur like, defects can be found at the later stage.
- It would be difficult to find out whether the defect arouse in interface or in module.

Big Bang...

Advantages:

- Convenient for small systems.

Disadvantages:

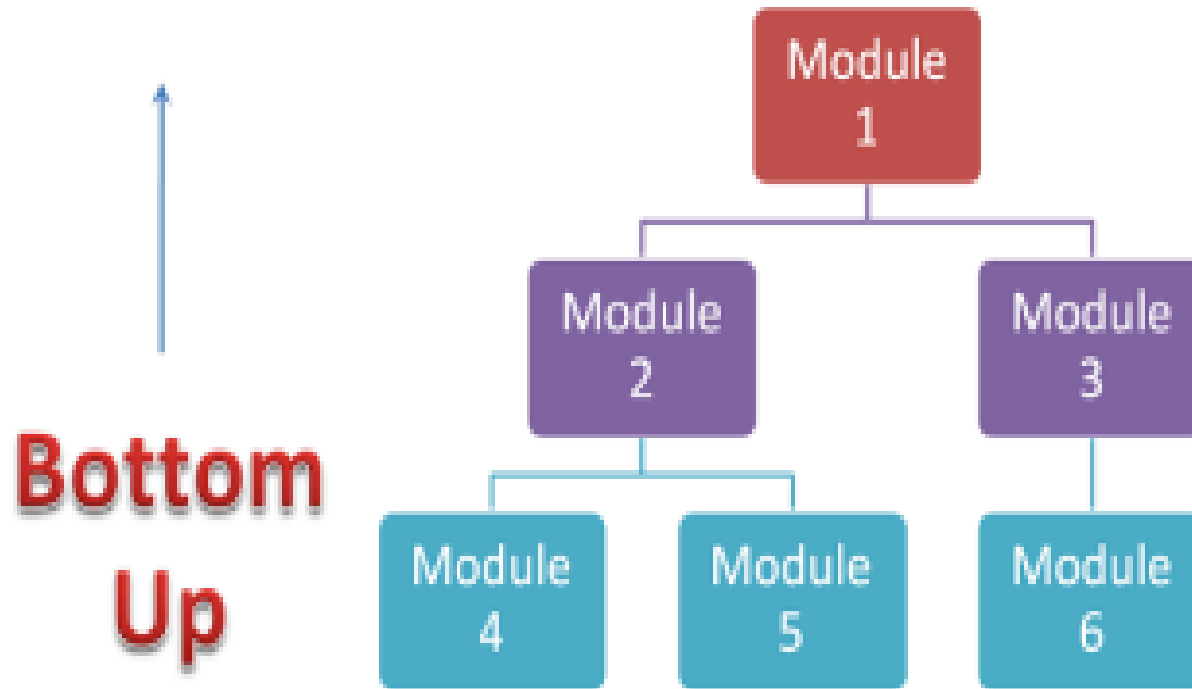
- Fault localization is difficult.
- Given the sheer number of interfaces that need to be tested in this approach, **some interface links to be tested could be missed easily**.
- Since the integration testing can commence only after "all" the modules are designed, testing team will have **less time** for execution in the testing phase.
- Since all modules are tested at once, **high risk critical modules are not isolated** and tested on **priority**. Peripheral modules which deal with user interfaces are also not isolated and tested on priority.

Incremental Approach

- In this approach, testing is done by **joining two or more modules** that are **logically related**. Then the other related modules are added and tested for the proper functioning.
- The process continues **until all of the modules are joined and tested successfully**.
- Incremental approach in turn is carried out by the following methods:
 - **Bottom up**
 - **Top down**
 - **Sandwich**

Bottom up Integration

- In the bottom up strategy, each module at **lower levels** is tested with higher modules until all modules are tested.



Bottom up...

Advantages:

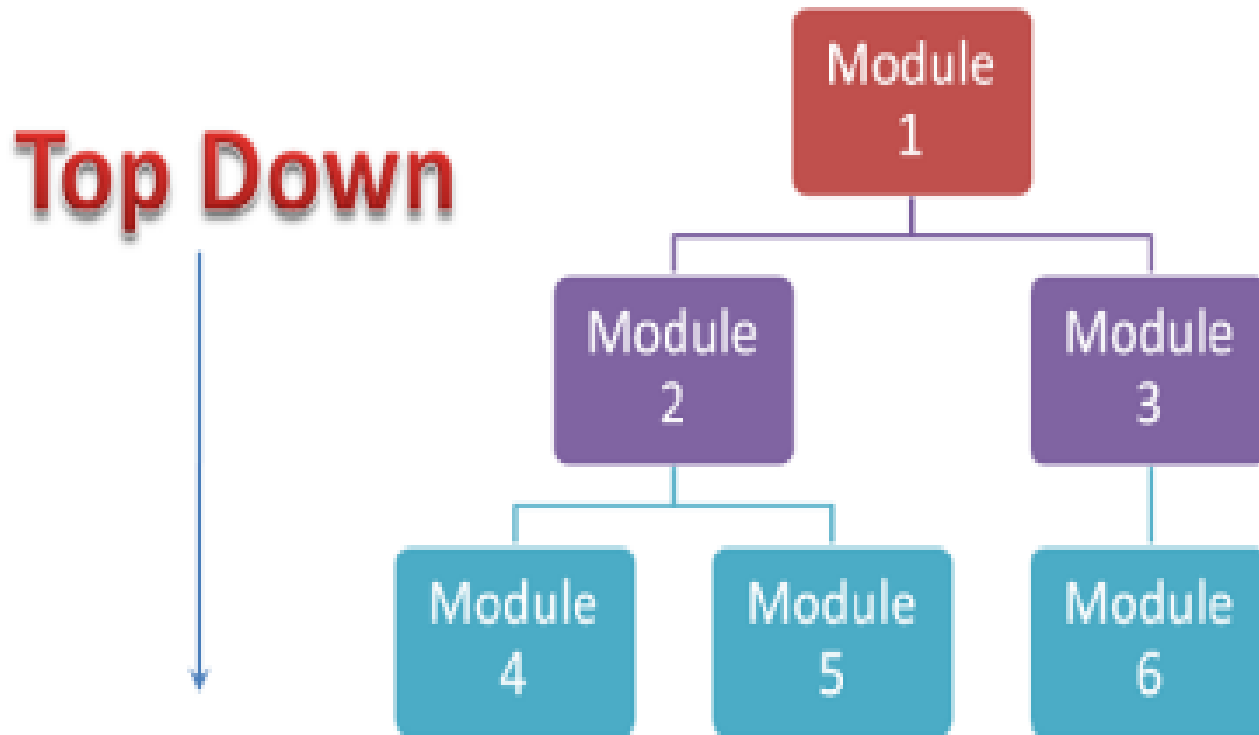
- Fault localization is easier.
- No time is wasted waiting for all modules to be developed unlike big-bang approach

Disadvantages:

- Critical modules (at the top level of software architecture) which control the flow of application are tested last and may be prone to defects.
- Early prototype is not possible

Top down Integration:

- In top-to-down approach, testing takes place from top to down following the control flow of the software system.



Top down...

Advantages:

- Fault localization is easier.
- Possibility to obtain an early prototype.
- Critical modules are tested on priority; major design flaws could be found and fixed first.
- Needs

Disadvantages:

- many stubs.
- Modules at lower level are tested inadequately.

Sandwich/Hybrid Integration Approach

- This approach is a **combination** of both top-down and bottom up integration testing.
- In this approach, the integration starts from **middle layer** and testing is carried out in **both directions** - towards higher level modules(upwards) and towards lower level module (downwards).
- This way includes benefits of both top-down and bottom up approach and helps in faster testing of the module's interfaces.

System Testing (End-to-End Testing)

- This is the first level of functional testing where the application is tested as a whole.
- The goal at this level is to evaluate systems compatibility with the outlined requirements and to see that it meets the anticipated quality standards.
- To verify the functional and technical specifications of the software, comprehensive testing is performed.
- The application is tested in an environment that is similar to the environment where the application will be deployed.
- Enables the testers to test, verify, and validate the business requirements as well as the application architecture.
- Performed by a knowledgeable and specialized testing team.

Users Acceptance Testing (UAT)

- Acceptance testing or user acceptance testing, is conducted by the **Quality Assurance (QA) Team** or **customers** to check whether the application meets the intended specifications and **satisfies clients' requirements**.
- Acceptance testing allows the tester to check for any **undetected bugs**, which can crash the system in future.
- Once acceptance testing is accomplished successfully, the application becomes **suitable for release**.
- The QA team must have a set of **pre-written scenarios** and **test cases** that will be used to test the application.

... UAT

- More ideas will be shared about the application and more tests can be performed on it to device its **accuracy** and the **reasons** why the project was initiated.
- Acceptance tests are not only intended to point out simple **spelling mistakes**, **cosmetic errors**, or **interface gaps**, but also to point out any bugs in the application that will result in **system crashes** or **major errors** in the application.
- By performing acceptance tests on an application, the testing team will reduce how the application will perform in production.
- There are also **legal** and **contractual** requirements for acceptance of the system.
- UAT includes:
 - a) **Alpha Testing**
 - b) **Beta Testing**
 - c) **Gamma Testing**

A) Alpha Testing

- Alpha testing is a type of acceptance testing performed to identify all possible issues/bugs **before releasing** the product to everyday users or public.
- It is performed **within the software company** or lab environment by **highly qualified specialists usually not by developers** of the product. **The testers are internal employees of the organization.**
- It can be referred to as a form of **internal acceptance testing**. Also in this stage, new functionality can be implemented in the software.
- The focus of this testing is **to simulate real users** by using black-box and white-box techniques.
- The aim is to carry out the tasks that a typical user might perform.
- This kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before Beta Testing.

Alpha ...

- During this phase, the following aspects will be tested:
 - Spelling mistakes
 - Broken links
 - Cloudy directions
 - The application will be tested on machines with the lowest specification to test loading times and any latency problems.

Alpha ...

Advantages:

- Provides better view about the reliability of the software at an early stage
- Helps simulate real time user behavior and environment.
- Detect many showstopper or serious errors
- Ability to provide early detection of errors with respect to design and functionality

Disadvantages:

- In-depth functionality cannot be tested as software is still under development stage
- Sometimes developers and testers are dissatisfied with the results of alpha testing

Beta Testing

- Beta testing is done **outside the software** developing **organization**, **not by its personnel**.
- Prerelease builds of the program are distributed to the end users who validate their functionality, reliability, usability and compatibility. It is known as a form of **external acceptance testing**.
- Beta testing lowers the risk of the product's failure and provides improved quality of the application by means of **customer feedback**.
- Direct feedback is a significant benefit of beta testing. This phase follows alpha testing.
- Beta testers often find a lot of defects of programs.

Beta...

- In this phase of software production, the development team bears no responsibility for the damage caused to the products when they are tried out by the clients or outsiders.
- No testing or lab environment is needed for beta tests.
- Beta testing companies engage a limited quantity of end users in the testing process to check the product in real world environment and decide if it is ready to be shipped to the customers.
- It aids partly to give the program a "real-world" test and partly to provide a preview of the next release.

Beta...

- In this phase, the audience tests the following:
 - Users install, run the application and send their feedback to the project team.
 - Typographical errors, confusing application flow and even crashes.
 - Getting the feedback, the project team can fix the problems before releasing the software to the actual users.
- The more issues you fix that solve real user problems, the higher the quality of your application will be.
- Having a higher-quality application when you release it to the general public will increase customer satisfaction.

Beta...

Advantages

- Reduces product failure risk via customer validation.
- It allows a company to test post-launch infrastructure.
- Improves product quality via customer feedback
- Cost effective compared to similar data gathering methods
- Creates goodwill with customers and increases customer satisfaction

Disadvantages

- **Test management** is an issue. As compared to other testing types which are usually executed inside a company in a controlled environment, beta testing is executed out in the real world where **you seldom have control**.
- **Finding the right beta users** and maintaining their participation could be a challenge

Types of Beta Testing

- There are different types of beta tests:
 1. **Traditional beta testing** - Product is distributed to the **target market**, and related data is gathered in all aspects. This data can be used for product improvement.
 2. **Public beta testing** - Product is publicly released to the **outside world** via online channels and data can be gathered from anyone. Based on feedback, product improvements can be done. For example, Microsoft conducted the largest of all Beta Tests for its OS -- Windows 8 before officially releasing it.

Types of Beta ...

3. Technical beta testing - Product is released to the internal group of an organization and gather feedback/data from the employees of the organization.

4. Focused beta testing- Product is released to the market for gathering feedback on specific features of the program. For example, important functionality of the software.

5. Post release beta testing - Product is released to the market and data is gathered to make improvements for the future release of the product.

Gamma Testing

- **Gamma testing** is a final stage of software testing before an appropriate public release, it is conducted primarily for safety purposes.
- It is intended to identify and fix minor defects that were found during beta testing.
- As a rule, this process is carried out with the involvement of as many end users as possible.
- In this final phase of software development, there is no need for going through all the in-house testing activities.
- Gamma tests can be run only when the application is ready for deployment to the specified requirements.

THE END
THANK YOU!!!

