Dev. Testing & Review

Akeem Heshimu, Michael Girardin, Aytac Guler

Software Development

Software development is the process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components.

Development of the software is an ongoing process to increase the quality of the end product.

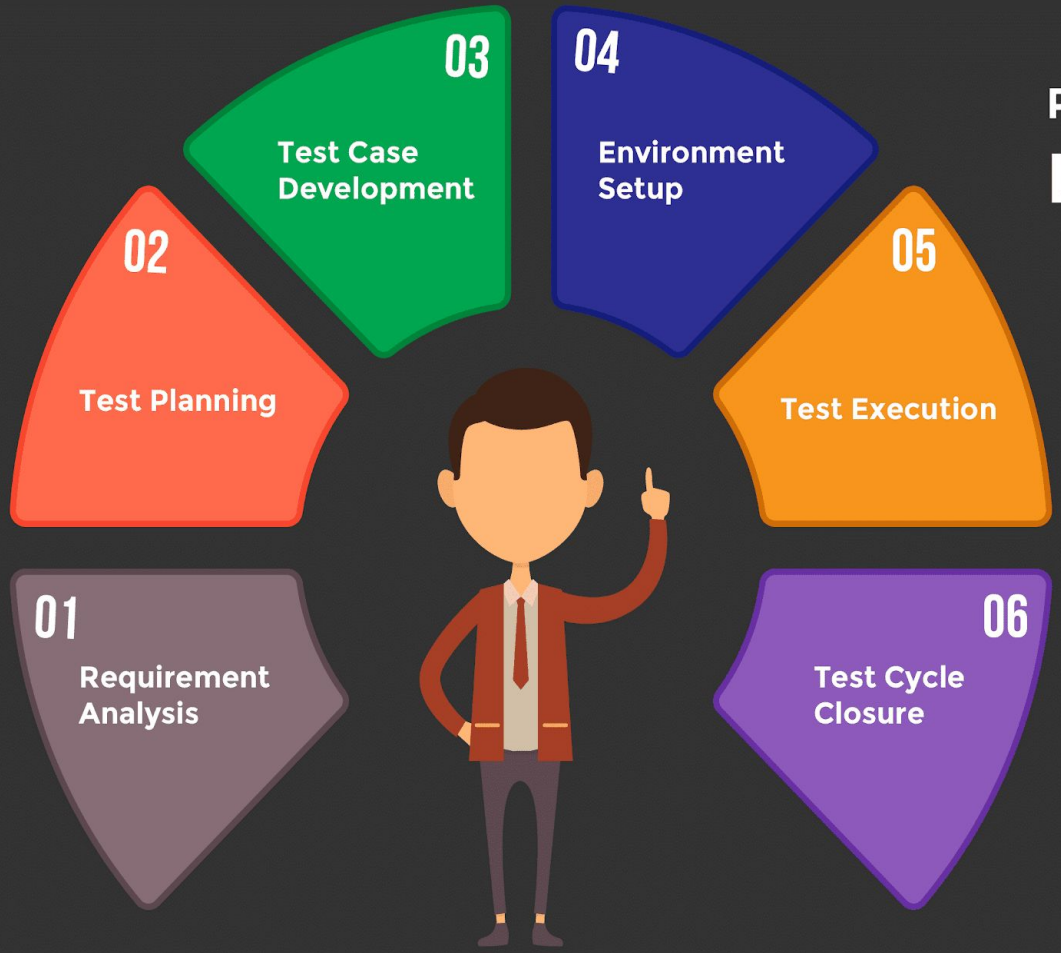
Testing Development Process

So a robust and stable software product can be delivered. More importantly, it is required to ensure whether they meet the specified requirements.

The process of executing a software under positive and negative conditions by automated means checks for the;

- Specification
- Functionality
- Performance

Product Testing
Life Cycle



Errors, Bugs, Fault & Failure

Error is a human action that produces the incorrect result that produces a fault.

Bug is the presence of error at the time of execution of the software.

Fault is the state of software caused by an error.

Failure is the deviation of the software from its expected results.

Objectives of Development Testing

- Uncover as many error/bugs as possible
- Demonstrate that the software meets its requirements
- Validate the quality of a software
- Generate high quality test cases, perform effective tests
- Issue correct and helpful problem reports

Testing Types and Methods

Manual Testing:
Testing the software to check for any bugs or unexpected behavior. Manual Testing has different stages such as, unit testing, integration testing, system testing, and user acceptance testing.

Automation Testing:
Repeating a manual test over again but will run more quickly. Automation increases the test coverage. Improving accuracy, saving time and money, in comparison to manual testing.

Black-Box Testing:
A person will test the software without having any knowledge about the software our how it works.

White-Box Testing:
Also called glass testing or open-box testing. A tester would need to know the aspect of the code.

Grey-Box Testing:
The tester has access to the database and the design documents allowing them to better proceed with creating a strategic test plan.

The two main levels to software testing are;

1. **Functional Testing**
2. **Non-Functional Testing**

The first 5 steps in **functional testing** are;

Steps	Description
1	The determination of the functionality that the intended application is meant to perform.
2	The creation of test data based on the specifications of the application.
3	The output based on the test data and the specifications of the application.
4	The writing of test scenarios and the execution of test cases.
5	The comparison of actual and expected results based on the executed test cases.

Functional Testing

Incorporates all test types designed to guarantee each part of software behaves as expected.

Unit testing: Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is different from the test data of the quality assurance team.

System Testing: System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards. This type of testing is performed by a specialized testing team.

Integration Testing: Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. integration testing can be done in two ways;

Step	Integration Testing Method
1	**Bottom-up integration** This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.
2	**Top-down integration** In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter.

Functional Testing Continued..

Regression Testing: When completing a regression test we make sure that any changes made, such as bug fixes haven't malfunctioned. It helps to;

- Minimize the gaps in testing when an application with changes made has to be tested.
- Testing the new changes to verify that the changes made did not affect any other area of the application.
- Mitigates risks when regression testing is performed on the application.
- Test coverage is increased without compromising timelines.
- Increase speed to market the product.

Acceptance Testing: Compared to all the other testing types acceptance is the most important. It is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirement. The QA team will have a set of pre-written scenarios and test cases that will be used to test the application.

Alpha Test: Known as the first stage of testing. Unit testing, integration testing and system testing when combined together is known as alpha testing.

Beta Test: Performed after alpha test

Non-Functional Testing

Incorporates all test types focused on the operational aspects.

Performance Testing: Helps to determine how an application will behave under various conditions.Performance testing includes;

- Load Testing
- Stress Testing
- Endurance Testing
- Spike Testing

Security Testing: Helps to determine if the information and data in a system is protected. Based on six basic principles of security;

- Integrity
- Confidentiality
- Authentication
- Authorization
- Availability
- Non-repudiation

Non-Functional Testing Continued..

Usability Testing: Measures an application's ease-of-use from the end-user perspective and is often performed together during the system or acceptance testing stages. The goal is;

- Determine if design meets the intended workflow
- Review the system or separate functionalities within the system.

Compatibility Testing: Determines how an application will work in different environments. Different environments consists of;

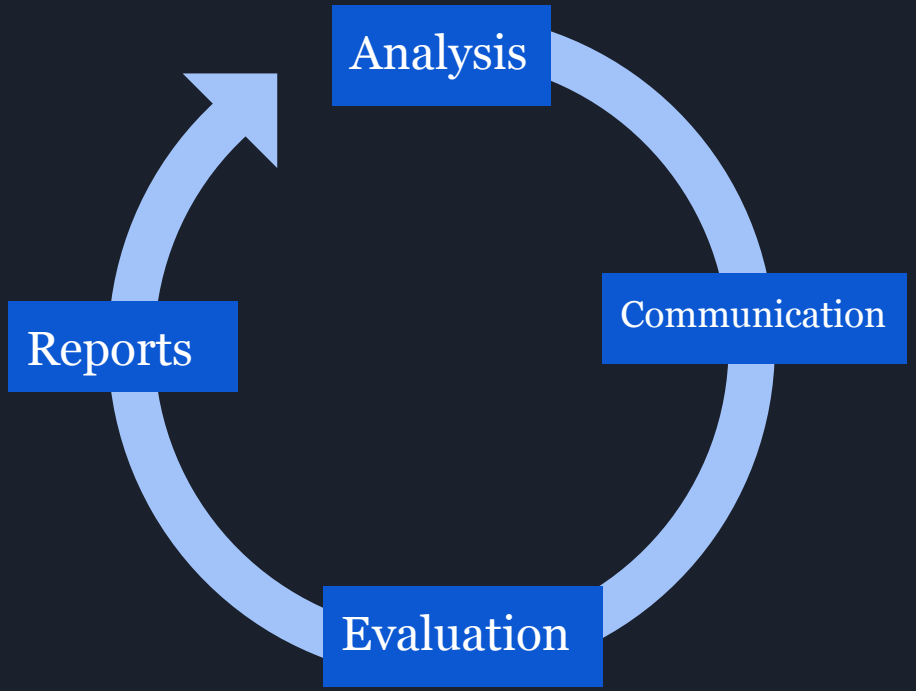
- Operating systems
- Platforms
- Browsers
- Resolution Configurations

Generating Reports

The software not only needs to meet all functions and reach a level of performance, but it must also be free of errors or “bugs”. Usually you do a quick smoke test to make sure nothing was broken during the fix phase.

Common communication goals;

1. Notify
2. Enlighten
3. Influence



Software Review

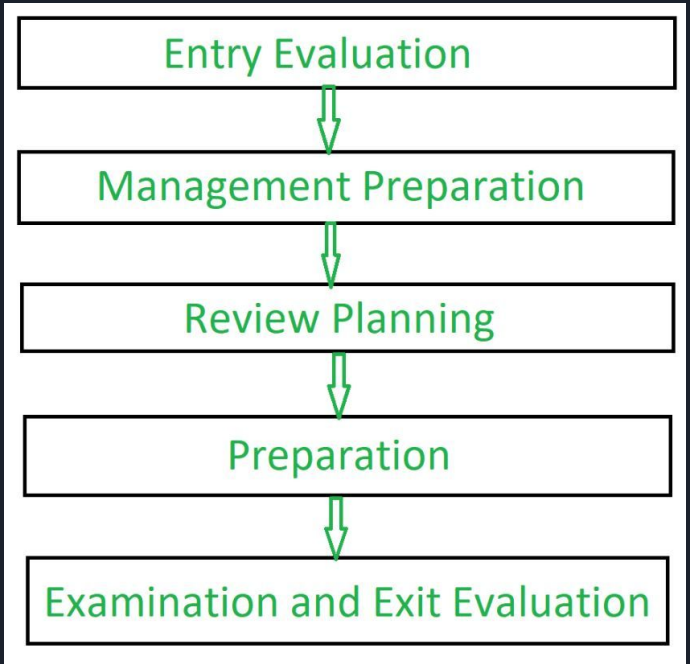
Software review is important to develop “best practices” for “industrial strength.”

- Improves not only the productivity of the code but also the team
- Fewer defects in the final software
- The costs are lower than all future stages as it helps to reduce reworks

Families of Review Methods

Method Family	Typical Goals	Typical Attributes
Walkthroughs	Minimal overhead Developer training Quick turnaround	Little/no preparation No formal process No measurement
Technical Reviews	Defect discovery Ambiguity resolution Training	Some formal process Multiple stages Wide range of discussion
Inspections	Detect and remove all defects efficiently and effectively.	Very formal process Measurement Verification

Review Process



Importance of Review

While working on a project you must verify that you have met all required system designs and describe your test plans and test cases. Look to really push your software and try to make sure that if others use it it won't break.

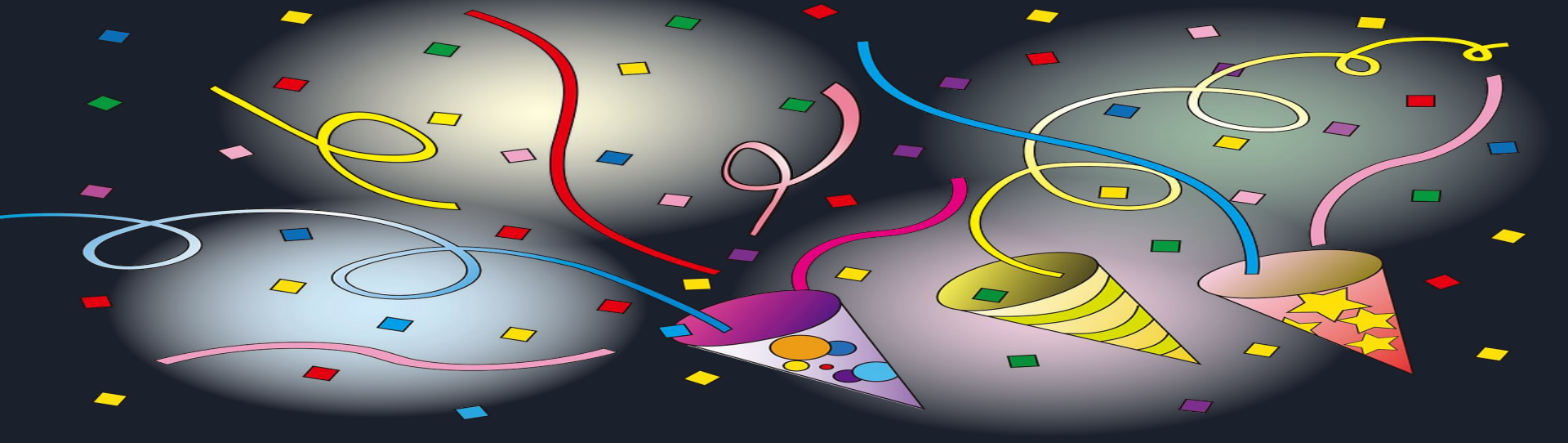
Any Questions

Question 1

The process of executing a software under different conditions checks to validate what 3 aspects of a software?

Answer

1. Specifications
2. Functionality
3. Performance



Question 2

What are the different methods of functional testing?

Answer

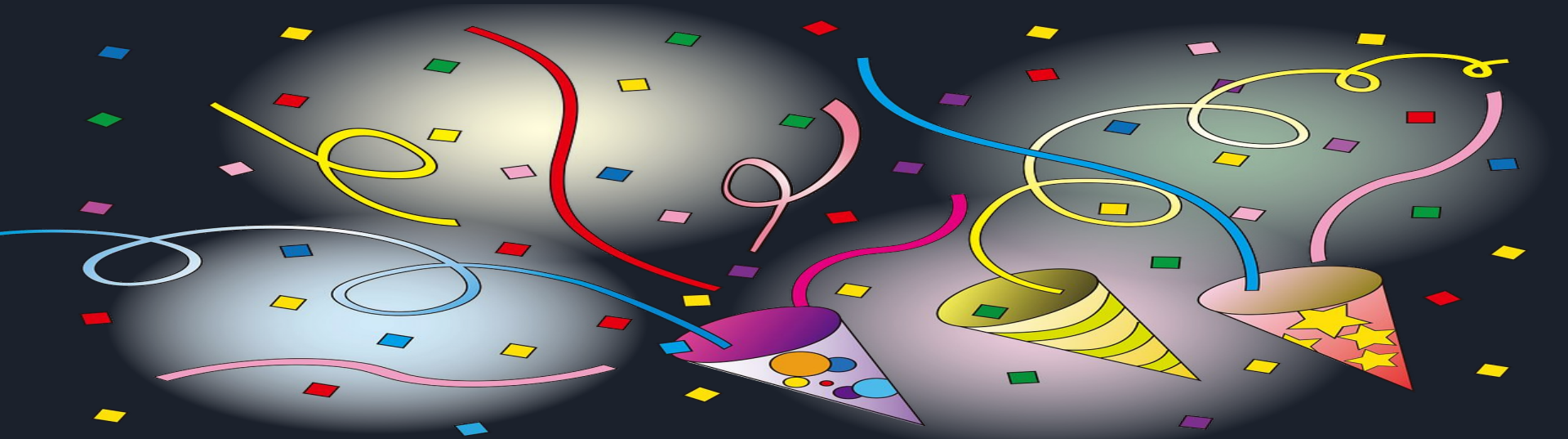
1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

Question 3

How can software reviewing improve your team and future software making process?

Answer

1. Analyzing past mistakes will prevent repeating the same mistakes
2. Team's software making process will become more efficient as they develop "best practices" to increase their "industrial strength"



If you didn't pay attention

We got you covered

SOFTWARE ENGINEER CHEAT SHEET

"It's a non-trivial amount of work"	=	I won't be able to copy and paste
"This is a temporary solution"	=	This is a permanent solution
"I'll be finished by Monday"	=	I won't tell you which Monday
"That's funny"	=	I literally hate you
"Do you want it done fast or do you want it done right?"	=	It will be neither
"Just file a bug"	=	I have no idea what you're saying right now
"That's an expansion of scope"	=	I forgot you needed that feature
"We can't test that"	=	We can but we won't
"I'll just put it behind a flag for now"	=	It will never ship
"I fixed the bug"	=	It started working again and I have no idea why
"I can't realistically estimate that"	=	Deadlines aren't my thing
"I don't think users will care about that"	=	I have dinner reservations at 8
"This is an Alpha release"	=	It doesn't work
"This is a Beta release"	=	We fixed a lot of bugs but it still doesn't work
"Let's phase 2 that"	=	I'm hoping you'll forget about it
"We are focusing on high priority fixes only"	=	I'm going to go take a nap
"We have a lot of technical debt"	=	I copied and pasted too much

Sources

https://www.tutorialspoint.com/software_testing/software_testing_levels.htm

https://www.guru99.com/software-testing-introduction-importance.html

https://www.softwaretestingmaterial.com/software-testing/

Additional Material

Software Testing Material