COURSE NAME – ADAPTIVE SOFTWARE ENGINEERING

COURSE CODE – 23CI2001

SESSION-23
TOPIC:
VALIDATION TESTING

AIM OF THE SESSION

To familiarize students with the basic concept of Validation Testing

INSTRUCTIONAL OBJECTIVES

This Session is designed to:

1. Demonstrate what is **validation testing**
2. Describe **importance of validation testing in software**
3. List out the phases of validation testing
4. Describe the types of validation testing

LEARNING OUTCOMES

At the end of this session, you should be able to:

1. Define validation testing
2. Importance of validation testing in software development

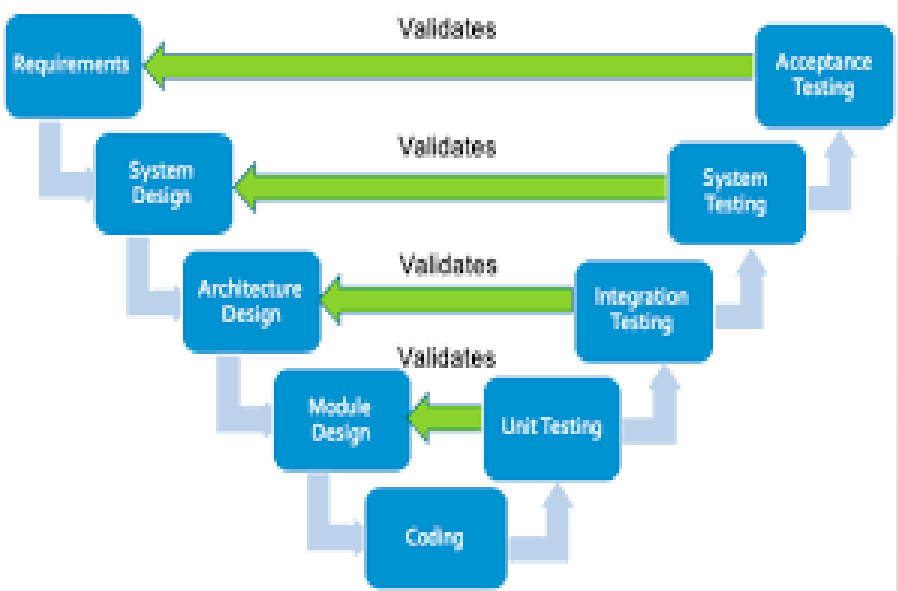
AGENDA

- What is validation testing
- Why validation testing is important
- Steps involved in validation testing
- Stages involved in validation testing
- Types of validation testing

VALIDATION TESTING

- Validation testing is the process of assessing a new software product to ensure that its performance matches consumer needs and expectations.
- Validation tests must be run after every feature or step in the development process is completed.

VALIDATION TESTING



WHY VALIDATION TESTING IS IMPORTANT

- Software validation can help product development teams ensure their product can satisfy customer needs and expectations.

  Validation testing can also help software developers identify and fix coding bugs or address other areas of improvement before launching the product.

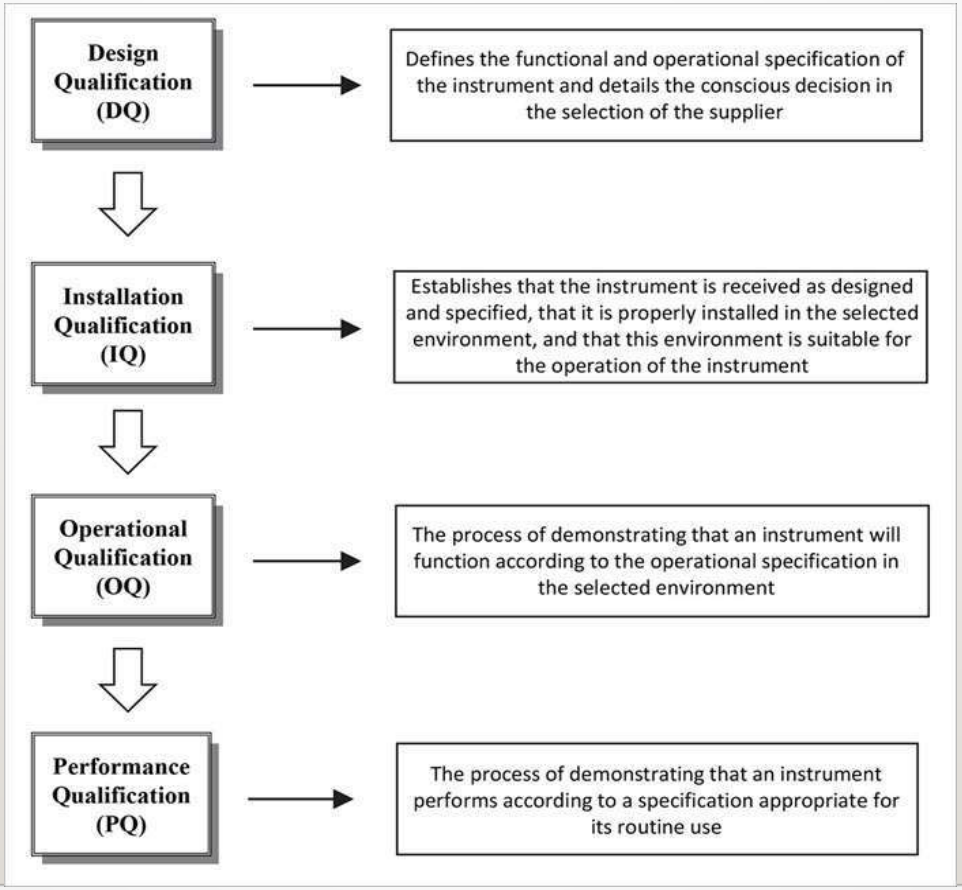
STEPS IN VALIDATION TESTING

- 1. You gather the business requirements for validation testing from the end user.
- 2. Prepare the business plan and send it for approval to the onsite/stakeholders involved.
- 3. On approval of the plan, you begin to write the necessary test cases and send them for approval.
- 4. Once approved you begin to complete testing with the required software, and environment and
- send the deliverables as requested by the client.
- 5. Upon approval of the deliverables, UAT testing is done by the client.
- 6. After that, the software goes into production.

STAGES IN VALIDATION TESTING

- 1.Design Qualification
- 2.Installation Qualification
- 3.Operational Qualification
- 4.Performance Qualification
- 5.Production

STAGES IN VALIDATION TESTING



DESIGN QUALIFICATION

- The process of design qualification, or DQ, includes creating a list of end-user business requirements and designing a validation testing plan to address them before launching the product. This plan can also be a useful written record of the design specifications that the developer and consumer desire. After writing the testing plan, development teams can seek approval from managers or shareholders before they begin the testing process.

INSTALLATION QUALIFICATION

- Installation qualification, or IQ, involves installing the software according to the validation testing plan. Product development teams may ensure that both system hardware and the installation process itself match the design specifications. This phase also involves ensuring that the test environment is suitable for product operation and matches the environment in which the product is likely to perform once the company releases it to the public.

OPERATIONAL QUALIFICATION

- Operational qualification, or OQ, involves testing the product with a variety of testing operations to ensure the product meets the specified user requirements.

PERFORMANCE QUALIFICATION

- Performance qualification, or PQ, testing verifies that a product can perform according to business needs in the real world. Developers on the internal team can perform alpha testing to assess the functionality of the software under simulated real-world conditions. After performing their own testing, a product development team can offer clients the chance to test the product through a process called beta testing.

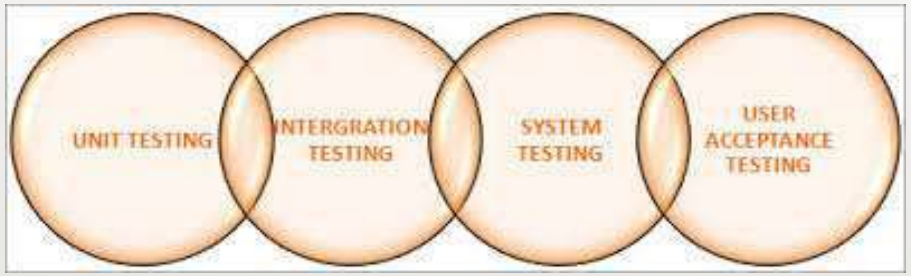
PRODUCTION

- After completing all levels of validation testing, a software product may go into production. This means that the product is ready to be marketed and sold to consumers. The software development team might help facilitate the deployment and installation process.

TYPES OF VALIDATION TESTING

- 1.Unit Testing
- 2.Integration Testing
- 3.System Testing
- 4.User Acceptance Testing(UAT)

TYPES OF VALIDATION TESTING



UNIT TESTING

- Unit testing is a form of validation testing that involves assessing small pieces of code individually. Units can include pieces of code like functions, methods, procedures, modules or objects. Testing these units separately from each other can help ensure that each is performing well. This improves the chance of the software functioning well overall.

INTEGRATION TESTING

- After performing unit testing to if each unit of code within the software is functioning correctly, software development teams can run integration testing to learn about how well the units function once they integrate them together into a larger system. Specifically, developers can ensure that data flow across modules is successful.
- The two main types of integration testing are the top-down approach and the bottom-up approach.

SYSTEM TESTING

- Also known as system-level testing or system-integration testing, this type of validation testing can assess the software as a complete system. This can help confirm that the product functions according to the end-to-end system specifications

USER ACCEPTANCE TESTING

- User acceptance testing, or UAT, is a form of performance qualification beta testing that invites a client to test the product and ensure that it meets their needs. This is often the last stage in the validation testing process and can be helpful because it tests the product in its proper environment. It can also help reveal challenges that the developers may not have noticed because they're already so familiar with the product.

SELF-ASSESSMENT QUESTIONS

1. Define Validation testing.
2. What is the importance of validation testing .
3. What is integration testing

REFERENCES FOR FURTHER LEARNING OF THE SESSION

TEXTBOOKS:

1. Roger S.Pressman, "Software Engineering – A Practitioner's Approach" 7th Edition, Mc Graw Hill,(2014).
2. Ian Sommerville, "Software Engineering", Tenth Edition, Pearson Education, (2015).
3. Agile Software Development Ecosystems, Jim Highsmith, Addison Wesley; ISBN: 0201760436; 1st edition

Reference Book

Agile Modelling: Effective Practices for Extreme Programming and the Unified Process Scott Amber John Wiley & Sons; ISBN: 0471202827; 1st edition.

WEB REFERNCES/MOOCS:

https://www.digite.com/kanban/what-is-kanban/
http://www.scaledagileframework.com
https://www.guru99.com/test-driven-development.html
https://junit.org/junit5/

THANK YOU

Team – ADAPTIVE SOFTWARE ENGINEERING