Build:

1. As per the CI/CD process once the end user places the **code** in the respective **version control** systems like TFS, TFVC, GitLab, Azure Devops, GitHub etc. **agent** or **runner** will pick the latest code into the **temporary work space** and initiate the process for **compiling** the application based on the respective language like Dotnet, Java, UI, Salesforce and etc.
2. Once the application language is identified we have to choose the supported build engine like **MSBuild** (For traditional), **Dotnet.exe** (For cloud based or core), ANT, MVN, NG, NPM, GRUNT, **GRUDLE, YARN, SFPX** etc.
3. The respected build engine will help us to compile the applications based on the properties that we have defined at the time of execution. If execution completed successfully we will get the artifacts/output/binaries. If execution failed then we have to validate following things.

    a. We have to validate all the required softwares are properly installed or not in the agent machine. If not we have to configure required softwares in the agent machine.
    b. If all the softwares are in place then we have to validate all required application dependencies/references/packages. If the dependencies are not restored properly then we have to restore them with the help of following tools based on the language it was built
        - NPM, Nuget, Dotnet, Yarn, Maven
    c. If Agent capabilities and its application dependencies are in place then we have to validate the compilation using respected build engine through command line. If application getting compilation success then we have to run the CI/CD process from the scratch. If compilation getting failed then we have to inform corresponding application development team through email and get them fix the issues on priority. Once the developer fixes the code issues he will place the latest code in version control and inform to devops engineer.
4. Once the latest code is available in the version control our build agent will start the process once pulls the latest code from version control into temporary work space and it will start the process for the compilation using respected build engine. If the application is getting compiled successfully without any issues we will proceed further with next stage like unit testing.
5. This process continues until compilation getting successful in case of any failures occur

Unit Testing:

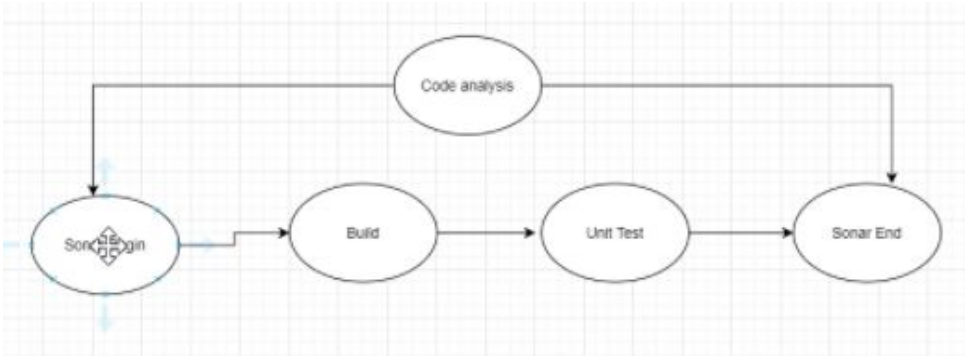
1. As part of the unit testing we are supposed to execute the unit test cases with help of respected test engines like MTM, NUnit, JUnit, MAVEN, DotNet.exe, NG, NPM, KARMA, JASMINE and so on. We are primarily responsible for the unit test cases execution but not for the results. As results always belongs to end user (Developer). As test cases results always belongs to corresponding application team.

2. If test execution is getting failed then we have to validate and fix it. The validation we are supposed to handle software capabilities and the dependencies in the agent machine.
3. Once the unit test cases execution is successful then we move to the further stage called **packaging.**

Packaging:

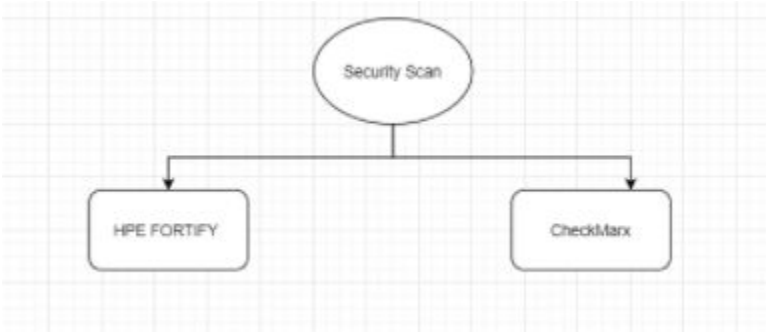
1. As part of packaging we are supposed to package the artifacts which are generated as part of the build stage in the format of .msi, .exe, .zip, .jar, .war, .octopack, .nupkg
2. Once packaging completed then we have to publish the artifacts into published artifacts staging directory of version control.
3. These artifacts will help us at the time of deployment

Code Quality:

1. As part of code quality we have two stages:
    a. **Code Analysis**
        b. As part of code analysis we are going to use a tool called **SonarQube** to analyze the code. The SonarQube analysis always relaise on sonar begin and sonar end.
    b. 

        b. Sonar End will help us to analyze the application and it will generate the report and publish into SonarQube Dash Board. That report consists following things
            1. Vulnerabilities
            2. Bugs
            3. Duplications
            4. Code chunks
            5. Code smells
            6. Code coverage
        c. We are primarily responsible for the code analysis execution till the report generated in the SonarQube dash board but not for the results inside the report. As results always belongs to corresponding application team.
    c. Security Scan:

b. As part of the security scan we are supposed to perform the security scan on top of application raw source code with the help of following security tools like
    1. HPE FORTIFY (Licensed, rank 1)
    2. Check Marks (Open source, rank 5)

c. 

d.

b. We are primarily responsible for the executing until report (.fpr) generated but not results in the report. Once the report is generated, the corresponding application team will verify how many errors, critical errors, high errors, warnings. Based on the error information the development team will apply the fixes until zero errors.
c. If the report consists any errors security team will never allow application team to move their code into production environment. So we have to enforce application team to maintain error free report before moving their code into production.

The above all stages comes under to **CONTINUOUS INTEGRATION**.

Deployment: (CD)

1. As part of the deployment in the CI/CD process we have to **deliver or deploy** the respected **components** (web , windows and etc.) of the application into various environments. The environments like
    a. DIT : Daily Integration Testing
    b. SIT : System Integration Testing (Staging)
    c. PERF : Performance Testing
    d. UAT : User Acceptance Testing (Includes load, # of users, ex : IRCTC, BigBillion Day)
    e. PROD : Production

2. During the deployment Build Agent will be downloading the published artefacts from artefacts staging directory of version control into temporary work space in the agent machine.
3. Once the artefacts are in place we have to perform deployment tasks into the set of servers of the environment
4. We can perform the deployment with help of power shell scripting or individual tasks in the release pipeline
5. If the deployment getting failed then we need to fix the issue and make the deployment happen. Once the deployment is completed then we have to check with the development team whether all components are deployed properly or not. Once the dev team confirms about the deployment then we can move to the next stage, nothing but testing.

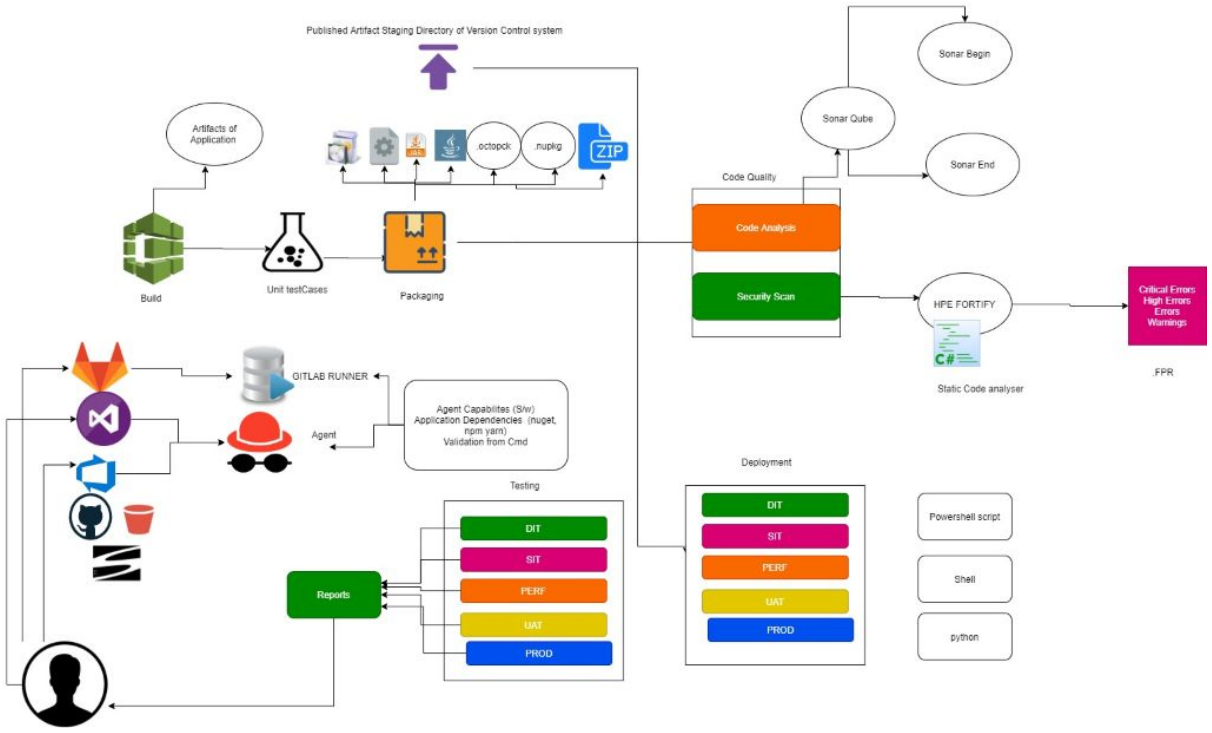
Testing :

1. Once we got the deployment into the respective environment then we have to perform automated test case execution against the respected deployed application. During the execution if anything is failed we have to validate required capabilities are configured in the agent machine. If not, we have to configure required software.
2. We are primarily responsible for executing the test cases but not for the results.
3. We have various types of test cases those are like
    a. BVT (Build Verification Testing) -> Runs against the DIT environment
    b. E2E (End to End Test cases)
    c. Large Test cases

Reports:

1. Once the deployment and testing stages are completed then we have received the results of respected environment verifications reports. These reports has to be sent to corresponding application team, nothing but end user.

The above three stages like Deployment, Testing and Reports comes under **CONTINUOUS DELIVERY**.



Points to remember:
End-to-end testing is a Software testing methodology to test an application flow from start to end. The purpose of this testing is to simulate the real user scenario and validate the system under test and its components for integration and data integrity.
It is performed from start to finish under real-world scenarios like communication of the application with hardware, network, database and other applications.

Large test cases will start from system requirements until end of test cases execution