TEST STRATEGY

Below is my general approach, covering all the possible steps to test the application during Software Devevlopment Life Cycle. Depending on the Methodologies (Agile/Waterfall) and project setup, the steps might vary.

Step 1 : Requirements Analysis

- Working closely with the Product Owner to understand requirements.
- Identifying functional and non-functional requirements.

Step 2 : Defining the time lines

- Estimating and defining the timelines, in case of Waterfall Methodologies, where-in the scope of requirements is huge.
- In case of Agile, timelines will be very short as the features to be tested will be small and also to support Continuous Integration and Continuous Deployment.

Step 3 : Preparing the test suites

- Preparing functional test suites as per the requirements defined.
- Preparing non-functional test suites as per the non-functional requirements.
    - Preparing Brower/Usability related test suite.
    - Preparing Performance testing test suite.
    - Preparing Load testing test suite.
    - Preparing security testing test suite.
- Preparing regression testing test suite.
- Preparing integration testing test suite.
- Preparing Requirements Traceability Matrix (RTM) and Coverage Matrix (CM).
- Tools like TestRail can be used to maintain test suites.

Step 4 : Test Execution

- Identifying the environment.
- Setting up the test data in Database, if necessary.
- Executing the test cases.
- Raising the defects in case of issues.
- Providing the below details in the defects raised.
    - Defect description
    - Steps to reproduce
    - Priority
    - Test data used
    - Environment details
    - Expected Results
    - Actual Results

- Verifying the defects fixed.
- Tools like JIRA can be used for defects management.

Step 5 : Integration Testing

- Performing integration testing when multiple systems are involved.

Step 6 : Regression Testing

- Performing the regression testing using the regression test cases.
- Raising the defects in case of any issues.

Step 7 : Continuos Integration – Continuous Deployment

- Performing testing parallel to development.
- Quick testing turnaround time.
- Concentrating more on testing the application and less in documentation.

Step 8 : Root Cause Analysis

- Performing Root Cause Analysis for the production issues.
- Identifying test scenarios and preparing test cases to avoid similar issues in further releases.
- Updating Regression suites with the identified test cases.

Step 9 : Learnings and Improvement

- Documenting learnings and sharing with the team.
- Improving the quality of testing by updating the test suites time-to-time, incase of production issues.