

CI & CD Pipeline

Venkat

From

Rise 'n' Shine Technologies

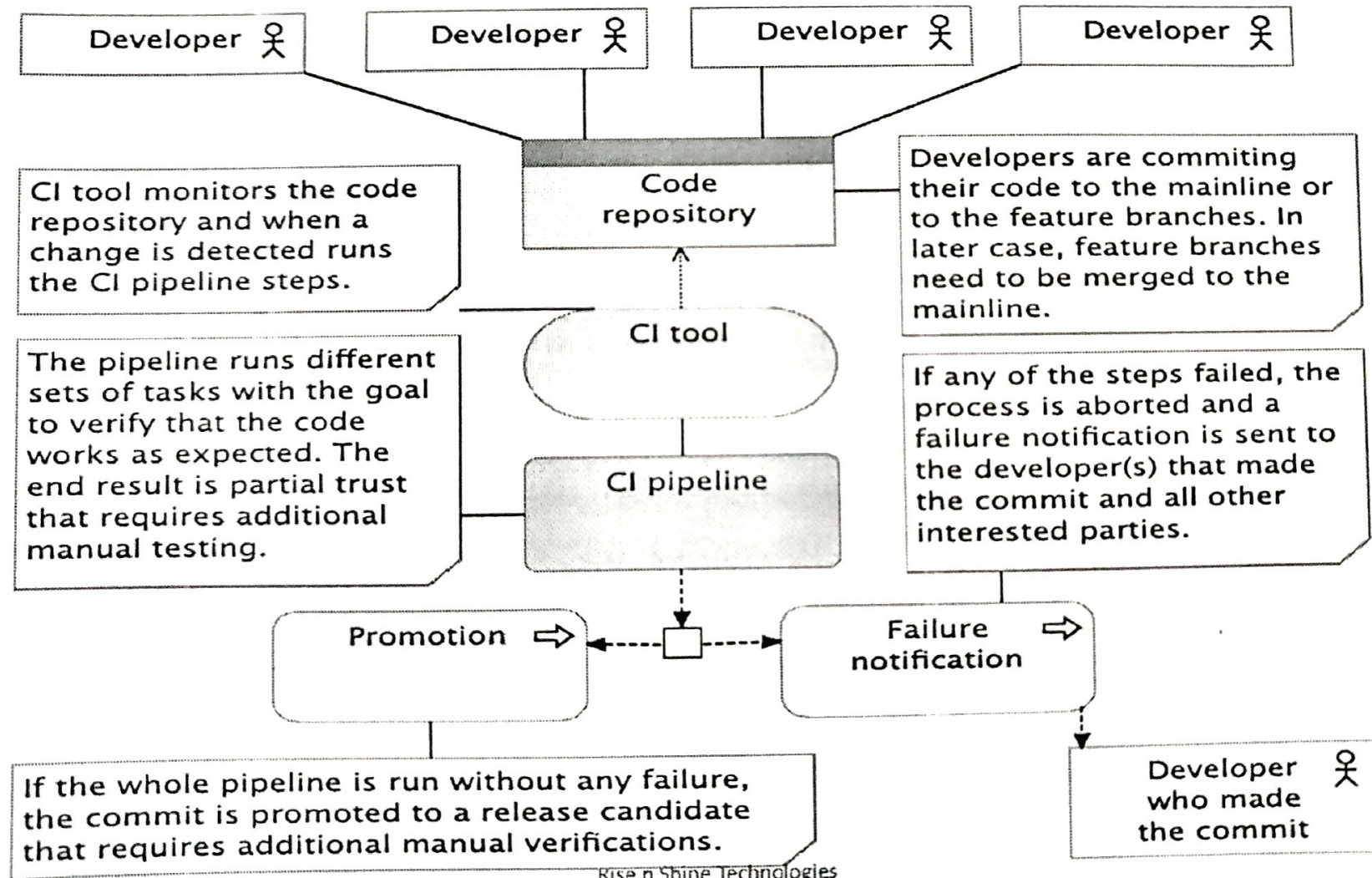
PipeLine

- The **most common flow (Pipeline)** is following.
 - 1. Pushing to the code repository
 - 2. Static analysis
 - 3. Pre-deployment testing
 - 4. Packaging and deployment to the test environment
 - 5. Post-deployment testing

1. Pushing to the Code Repository

- CI tools (Jenkins / Hudson) are monitoring the code repository, and whenever a commit is detected, the code is checked out (or cloned) and the CI pipeline is run.
- The pipeline itself consists of a set of automated tasks run in parallel or sequentially. The result of the pipeline is either a failure in one of its steps or a promotion.

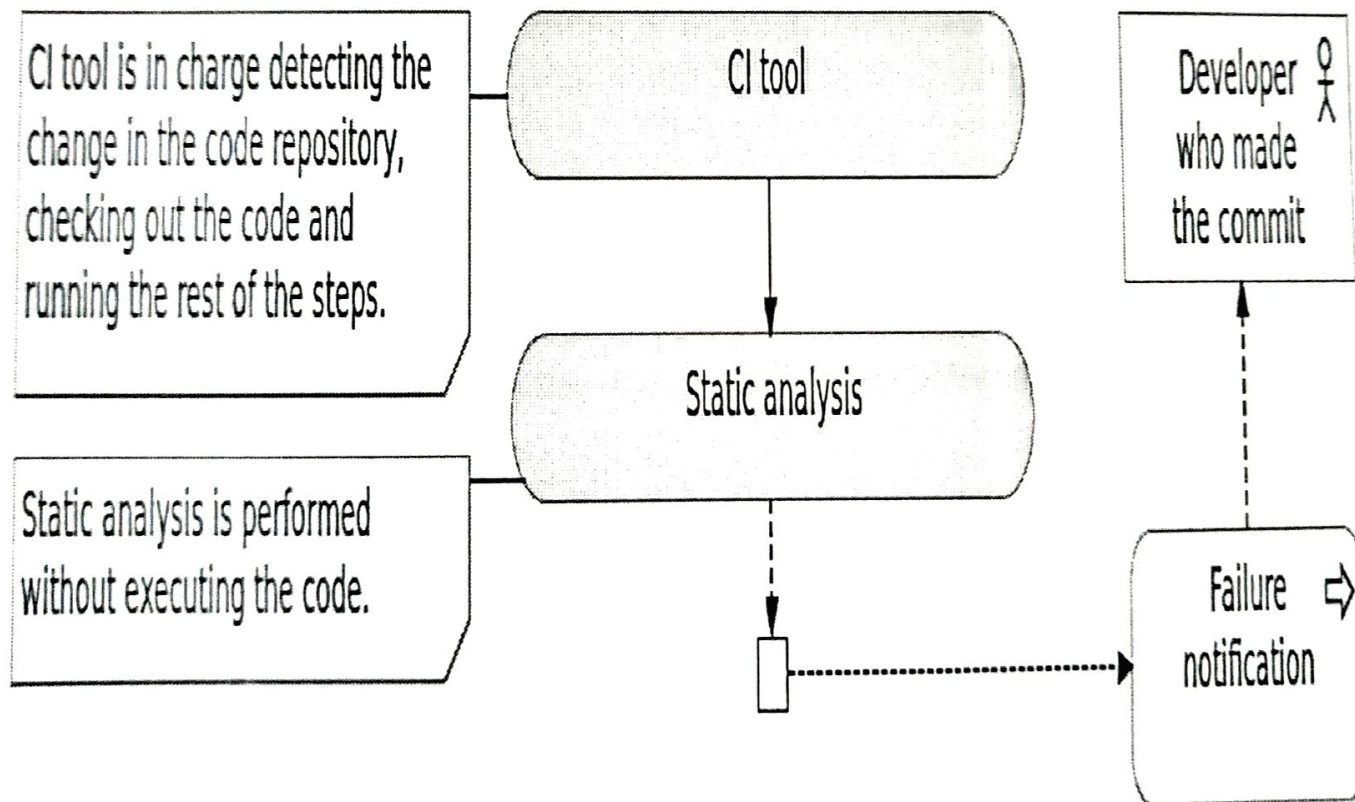
1. Pushing to the Code Repository



2. Static Analysis

- The *static analysis* goals vary from highlighting possible **coding errors** to make sure that agreed formatting is followed.
- CheckStyle and FindBugs for Java, JSLint and JSHint for JavaScript, and PMD for a variety of languages.
- Static analysis is often the first step in the pipeline for the simple reason that its execution tends to be very fast and in most cases faster than any other step we have in the pipeline.

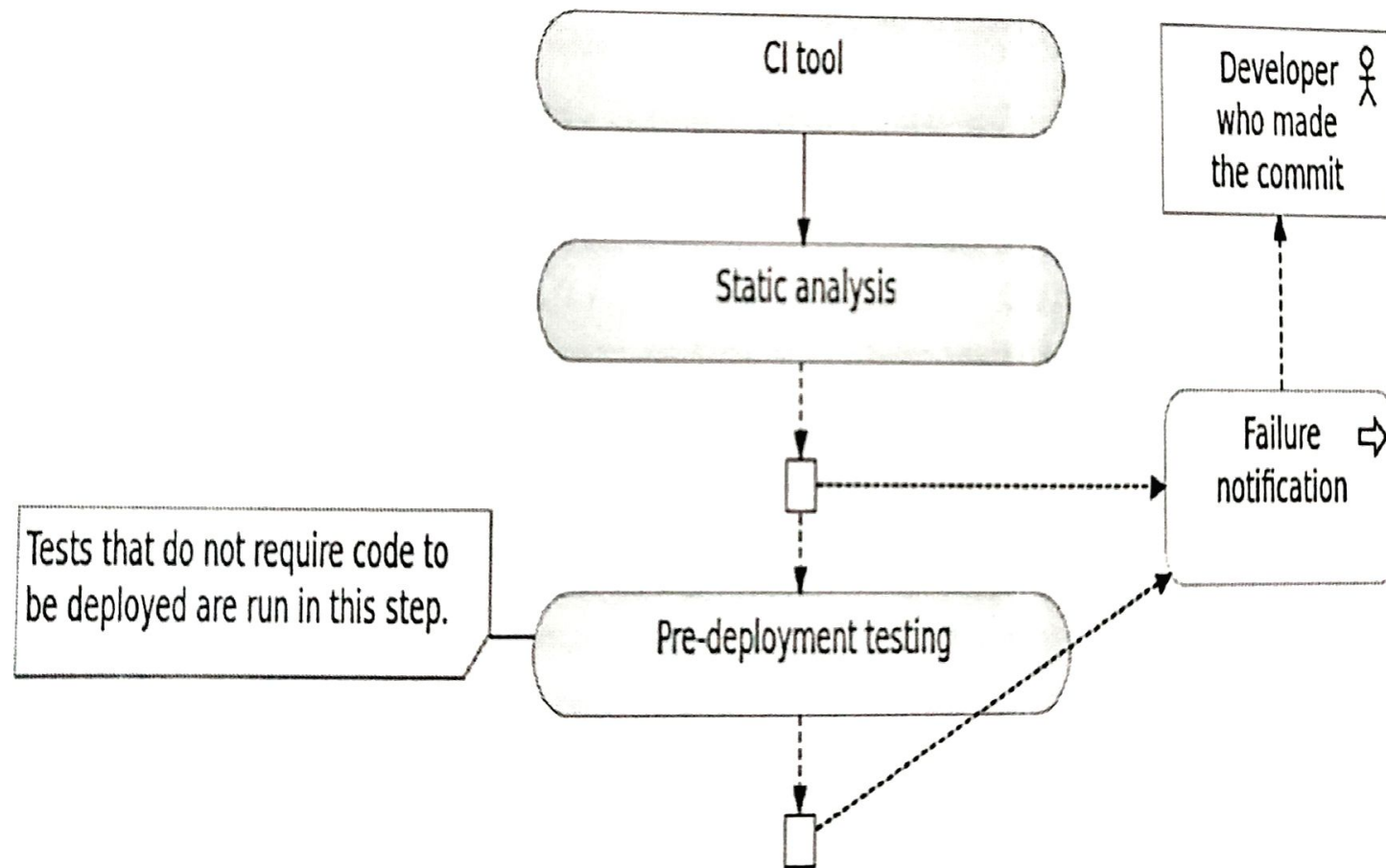
2. Static Analysis



3. Pre-Deployment Testing

- *pre-deployment* tests should be mandatory.
- *Unit tests* always fall into this category and with few others that might be run as well. If, for example, you can execute *functional tests* without deploying the code, run them now.

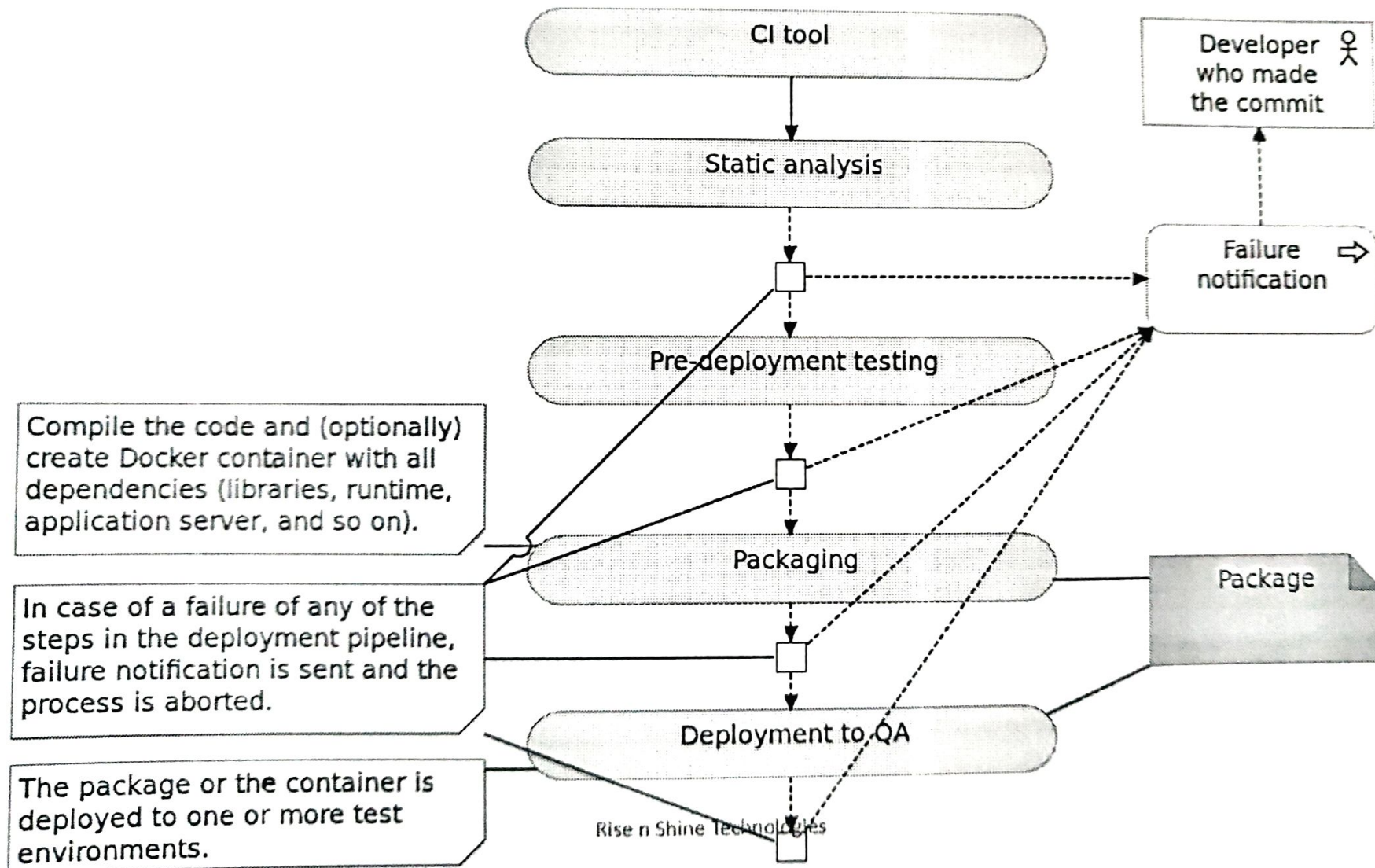
3. Pre-Deployment Testing



4. Packaging and Deployment to the Test Environment

- Once we did all types of verifications that could be done without actually deploying the application, it is time to package it.
- In the Java world we would create JAR or WAR or EAR files.
- Some programming languages do not require us to do anything in this phase except possibly compress all the files into a ZIP or TAR a file for easier transfer to servers.
- Once the deployment package is created, we can proceed to deploy it to a test environment.

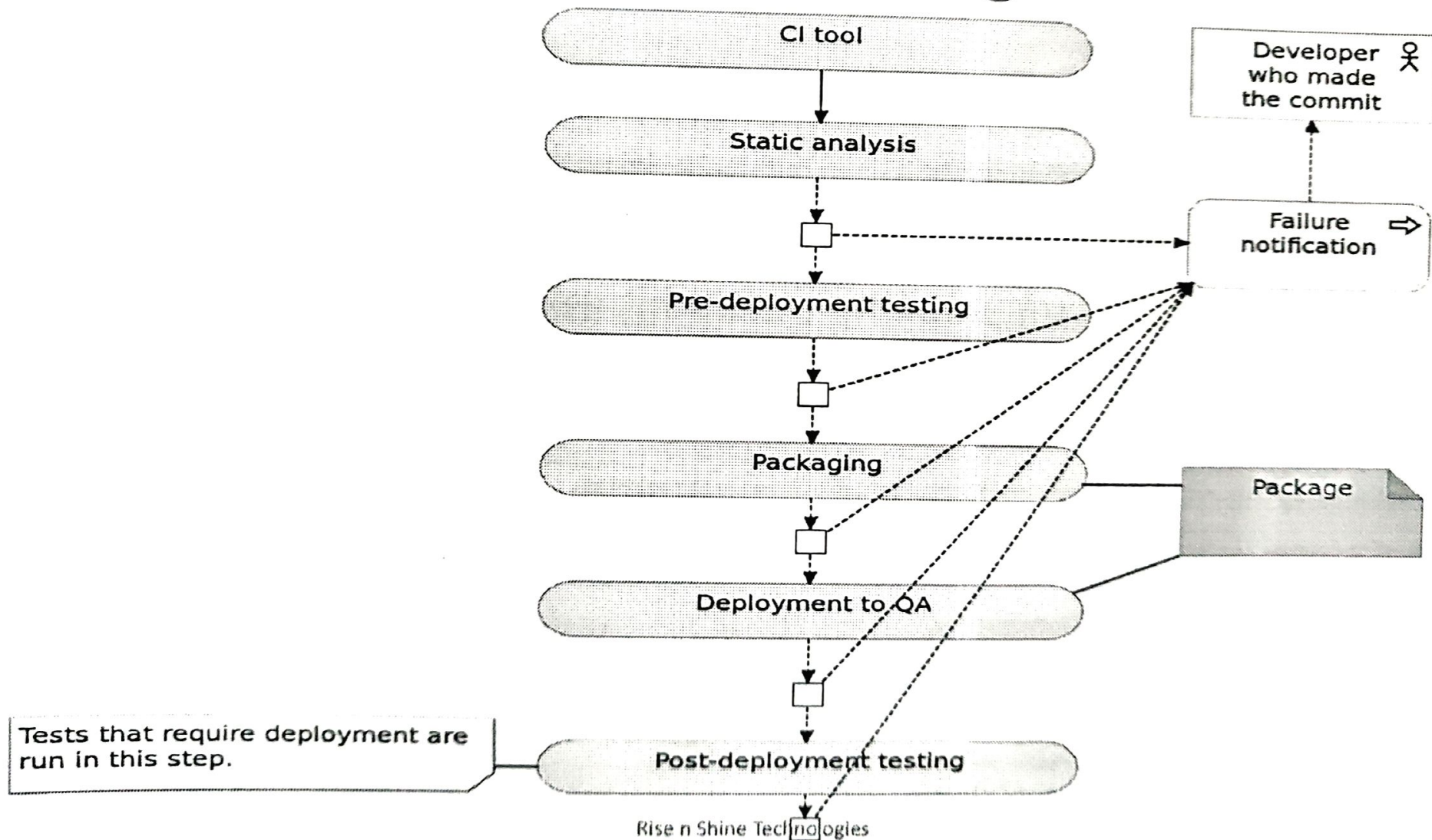
CI pipeline: packaging and deployment



5. Post-Deployment Testing

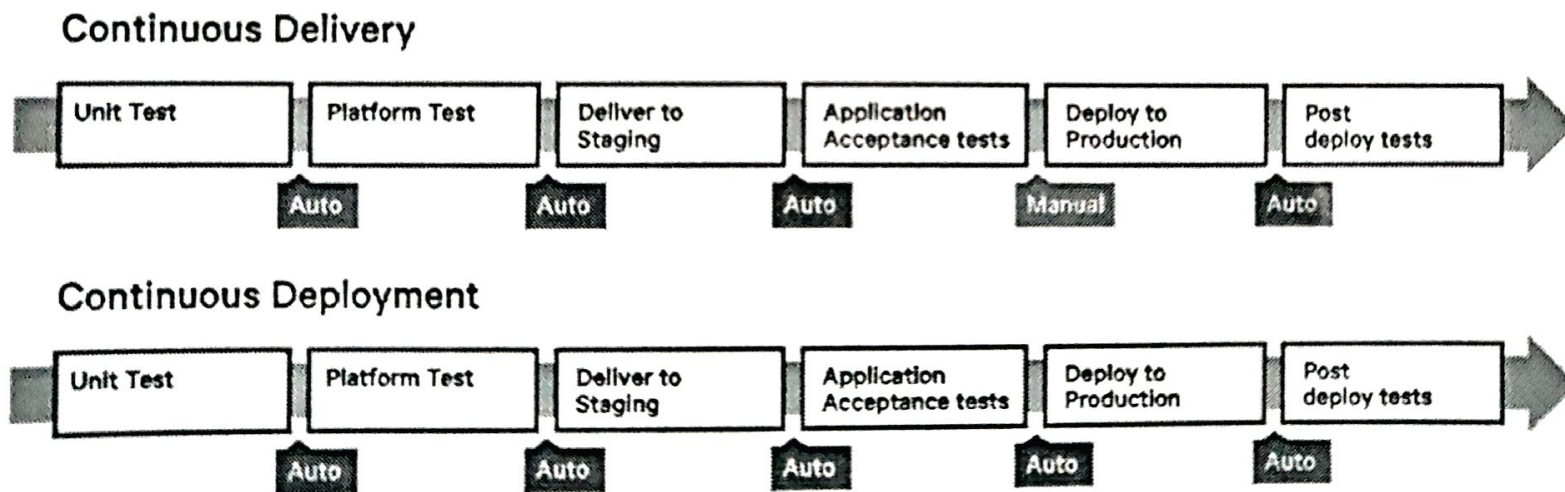
- Once deployed to a test environment, we can execute the rest of the tests; those that could not be run without deploying the application
- they include functional, integration and performance tests.
- Once the execution of post-deployment tests is finished successfully, the CI pipeline is typically completed as well. Packages or artifacts we generated during the *packaging* and *deployment to test environment* are waiting for further, usually manual, verifications.

CI pipeline: post-deployment testing

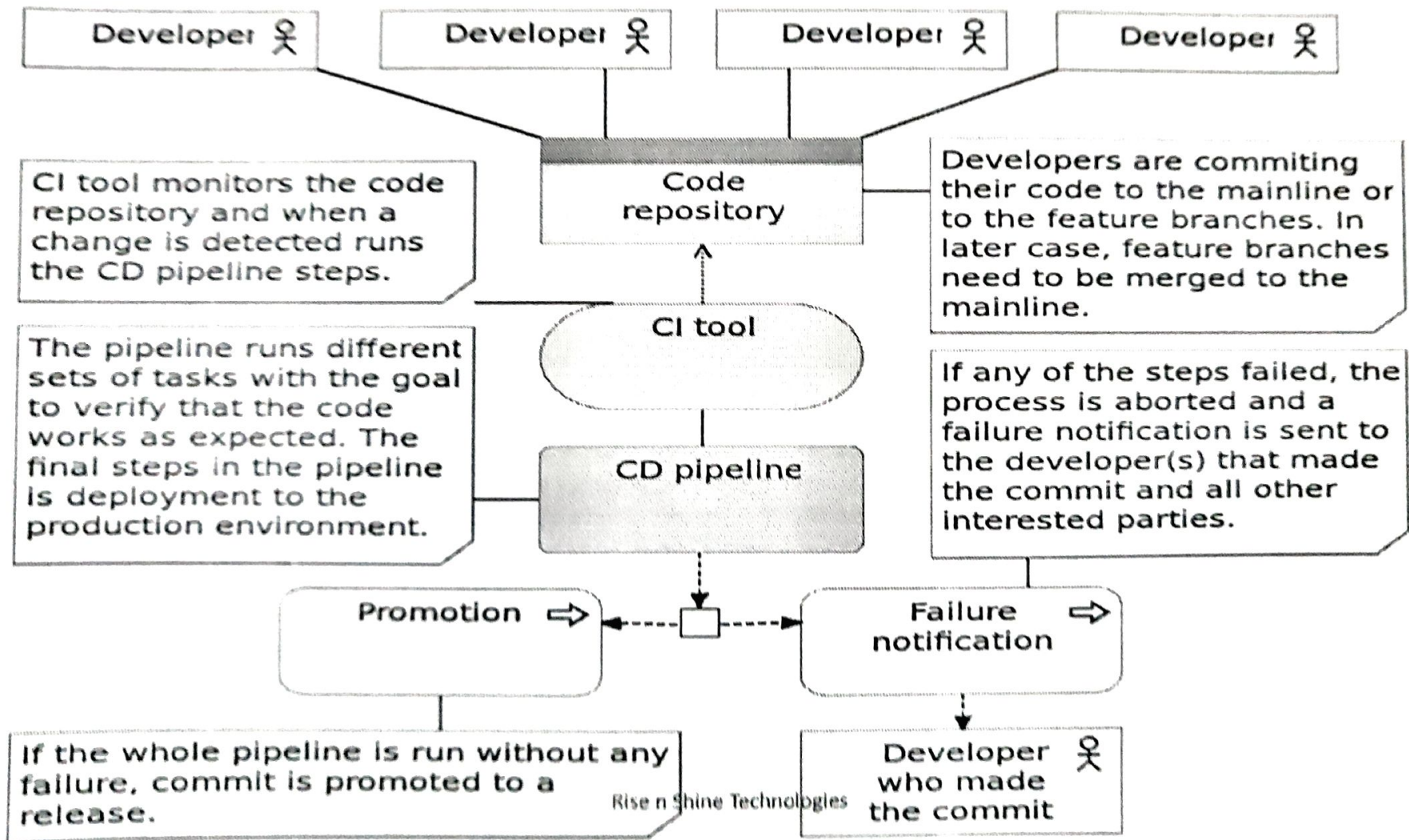


Continuous Delivery and Deployment

- The *continuous delivery* pipeline is in most cases the same as the one we would use for CI.
- In other words, every successful run of the pipeline *can be deployed to production*, no questions asked.



Continuous delivery process



Continuous deployment process

- The *Continuous deployment* pipeline goes a step further and automatically deploys every build that passed all verifications. It is a fully automated process that starts with a commit to the code repository and ends with the application or the service being *deployed to production*.

Continuous deployment pipeline

