Continuous Deployment – Workshop

HS Düsseldorf, 07.05.2019, PT-Support

Capgemini Standards

devonfw

- A standardized development platform including:
  - IDE
  - Architecture blueprints
  - Processes
  - …
- OpenSource (https://www.devonfw.com)
- Full sample application available (https://github.com/devonfw/my-thai-star)

PRODUCTION LINE

- A standardized “DevOps” Toolchain
- Not open-source

Capgemini Standards

devonfw

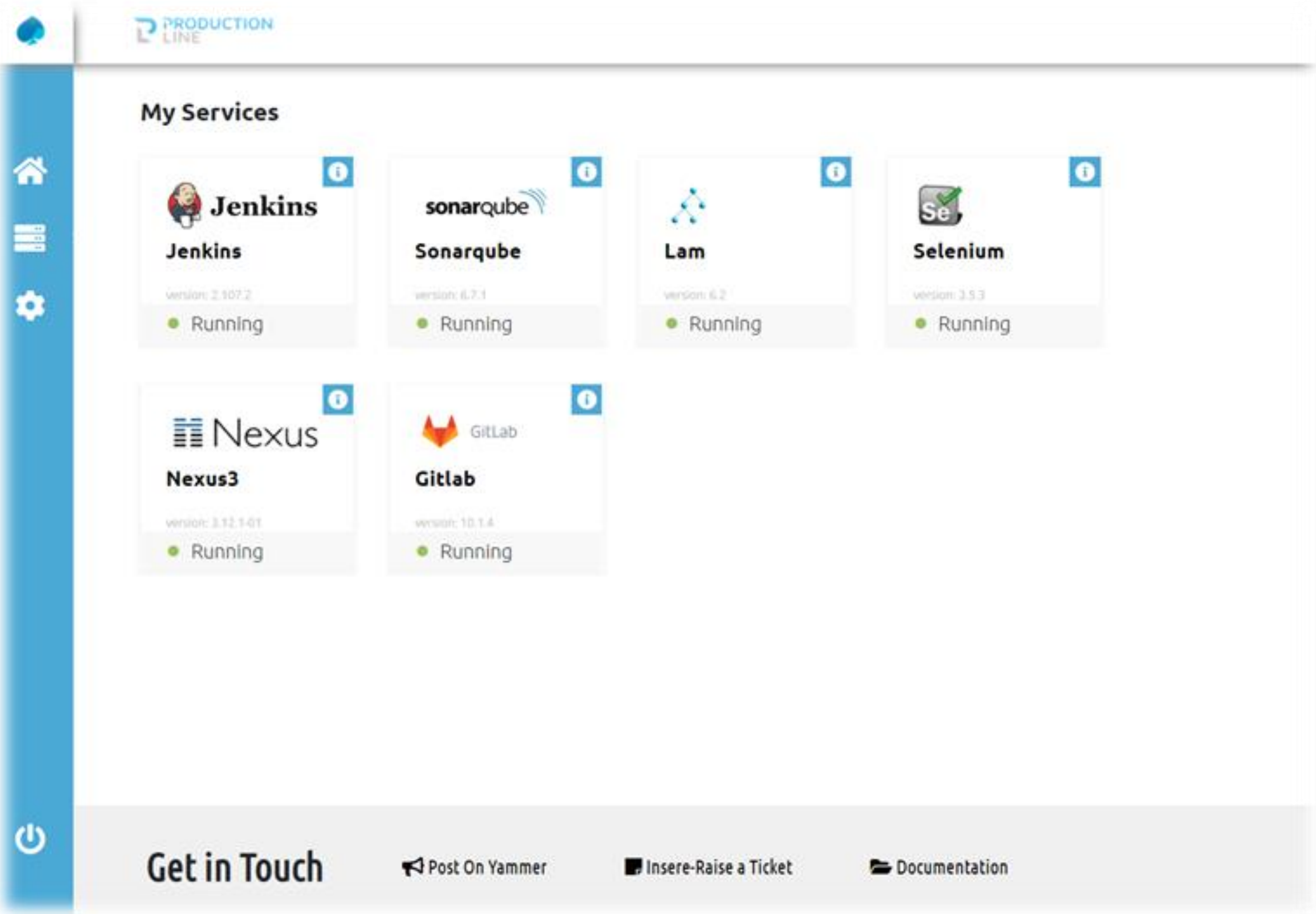
- A standardized development platform including:
  - IDE
  - Architecture blueprints
  - Processes
  - …
- OpenSource (https://www.devonfw.com)
- Full sample application available (https://github.com/devonfw/my-thai-star)

PRODUCTION LINE

- A standardized "DevOps" Toolchain
- Not open-source

https://mp.cd.ce.capgemini.com

Workshop Instance



My Services

Jenkins
version: 2.107.2
● Running

Sonarqube
version: 6.7.1
● Running

Lam
version: 6.2
● Running

Selenium
version: 3.5.3
● Running

Nexus3
version: 3.12.1-01
● Running

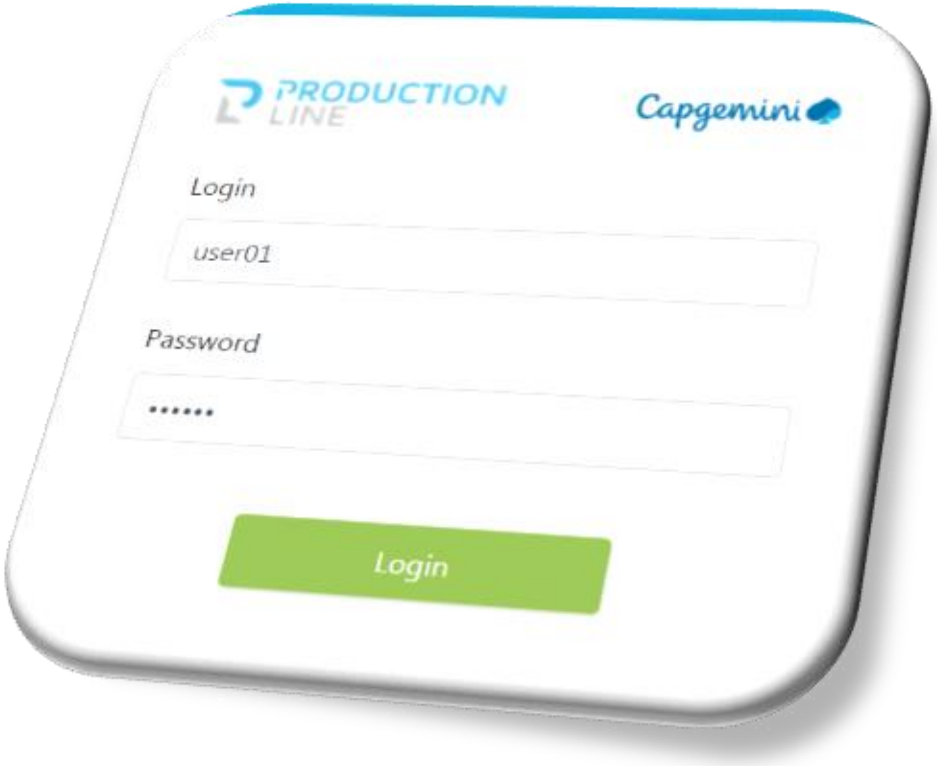
Gitlab
version: 10.1.4
● Running

Get in Touch

Login and General Rules

Go to **https://mp.cd.ce.capgemini.com**

- Login with username userXX and password userXX, where XX ranges from **02** to **25**
- Keep the following in mind:
  - The repositories of user01 are templates, so don’t edit them
  - If you login as userXX, e.g. user05 and there are already repositories for this user, it's highly likely that another person is using this account, so please try a different one
- At the end of everything, please delete your repository and your Jenkins-Jobs!



Login

user01

Password

••••••

Version Control Details

Click on the Gitlab icon, this will redirect you to Gitlab.

- There, you click on the + left from the search bar on the top and click on "New project"
- Then you pick "import project" -> "git Repo by URL" with the following settings:
  - URL https://user01:user01@mp.cd.ce.capgemini.com/gitlab/user01/my-thai-star-01-frontend.git
  - As "Project name" you choose "my-thai-star-**XX**-frontend”
  - As "Visibility Level" you pick "Public"
- Click on "Create project"

Repeat the process with the following URL https://user01:user01@mp.cd.ce.capgemini.com/gitlab/user01/my-thai-star-01-backend.git and the Project name "my-thai-star-**XX**-backend"

(**XX** = Your username-number!)

Build Details

Go back to https://mp.cd.ce.capgemini.com/ and click on Jenkins

- There you find a "JobCreation"-Job. Clock on The Clock with the green Arrow to the right
- After the job is done, you can click on the "userXX" tab or go to
- There you should have two jobs now. Start the Jobs

- While they are running, you can add a GitLab-Webhook.
  - For this Click on your "my-thai-star-01-backend-Java_master" Job, and the on the left you click on "Configure" - Under "Build Triggers", there is a checked checkbox "Build when a change is pushed to GitLab.“
  - Under advanced, you will find the "Secret Token" field. Click on generate and copy the Token.
  - Go to your GitLab backend repository and clock on "Setting"->"Integrations"
  - In the URL-field you put:
    - http://jenkins-core:8080/jenkins/project/my-thai-star-XX-backend-Java_master
    - http://jenkins-core:8080/jenkins/project/my-thai-star-XX-frontend-Angular_master
  - In the Secret Token field, you put the copied Secret Token. Clock in "Add webhook" and you are done.

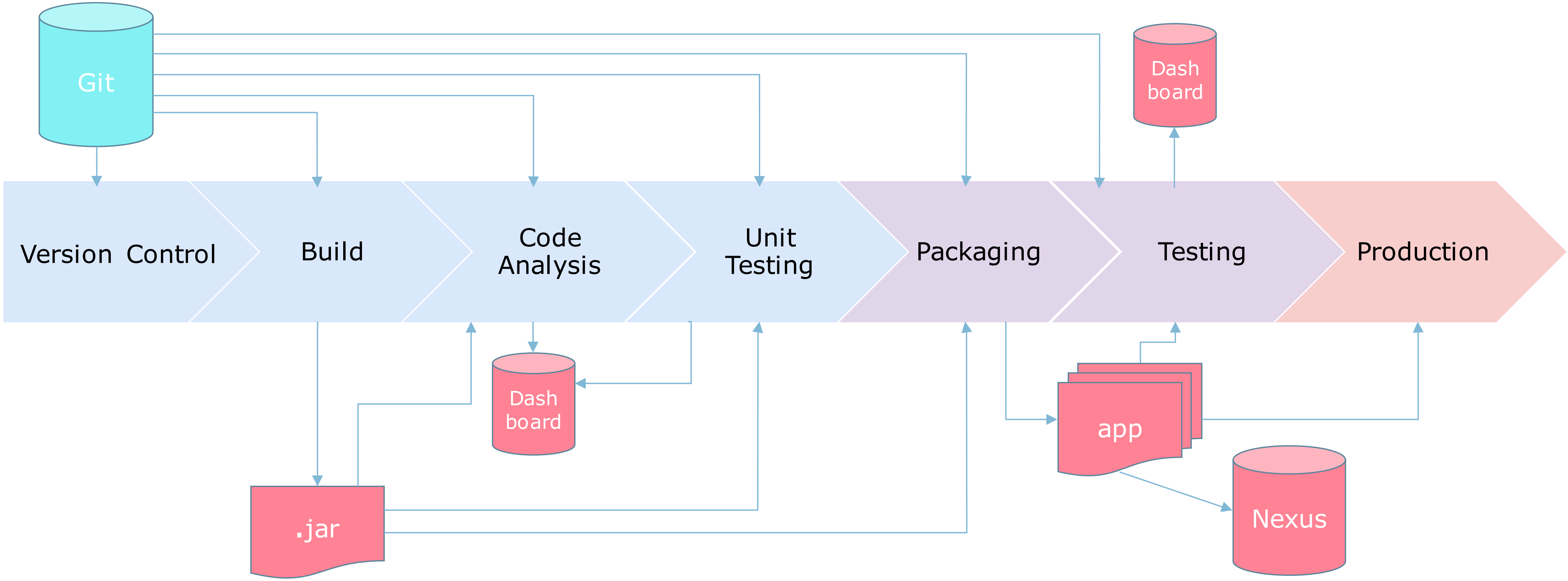
Exercise

Clone the repositories to your local machine (you should be able to do this on your own)

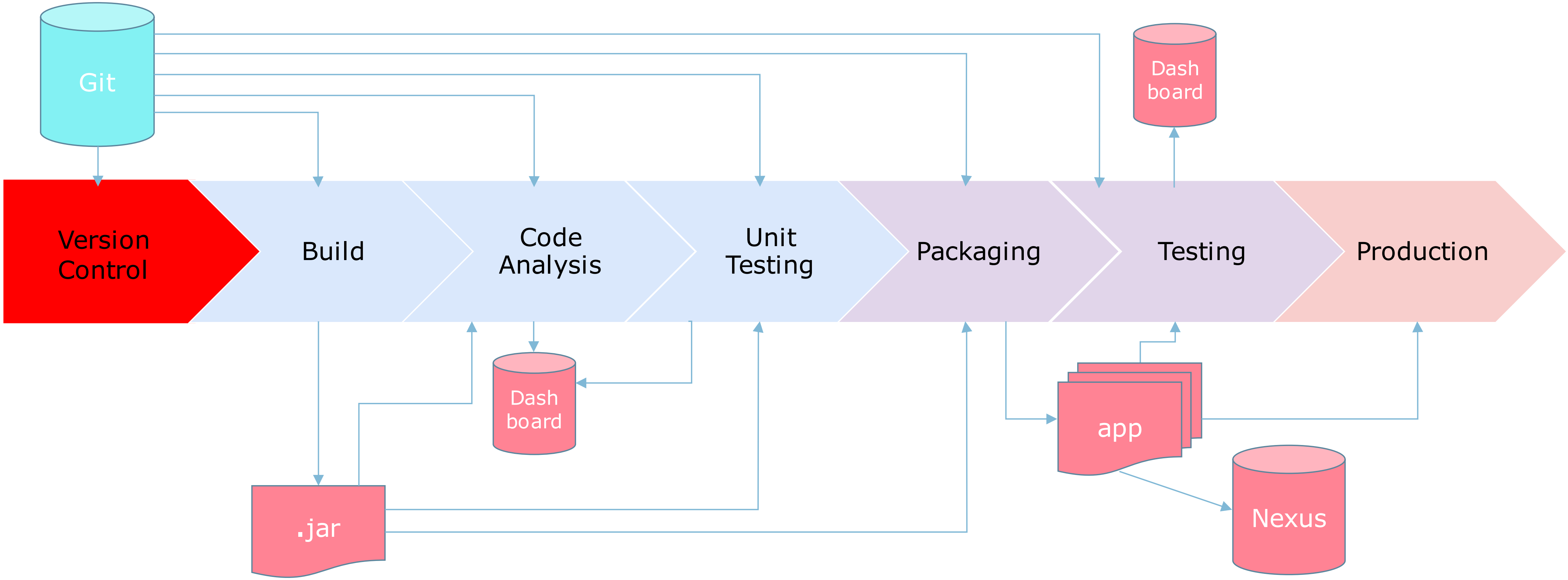
Go back to your Jenkins-Jobs. You will notice, that both jobs failed. Search the logs, why they failed and try to fix the issues (in the Code). You can see your SonarQube analysis reports by going back to https://mp.cd.ce.capgemini.com/ and click on SonarQube. There you type "userXX" in the searchBar.

Continuous Deployment Pipeline

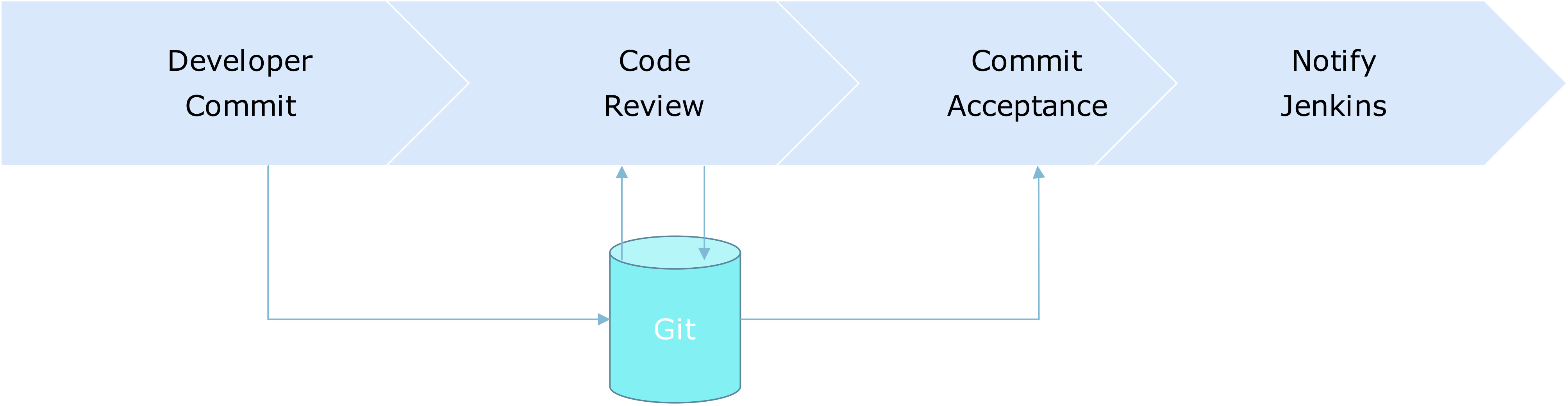
Continuous Deployment contains Continuous Integration, Continuous Delivery and adds Deployment to Production.

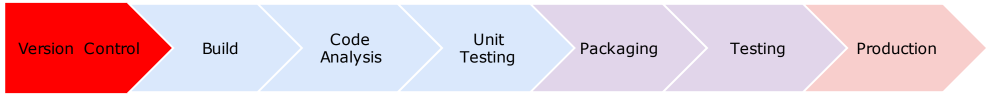


Version Control – Where the Developer begins

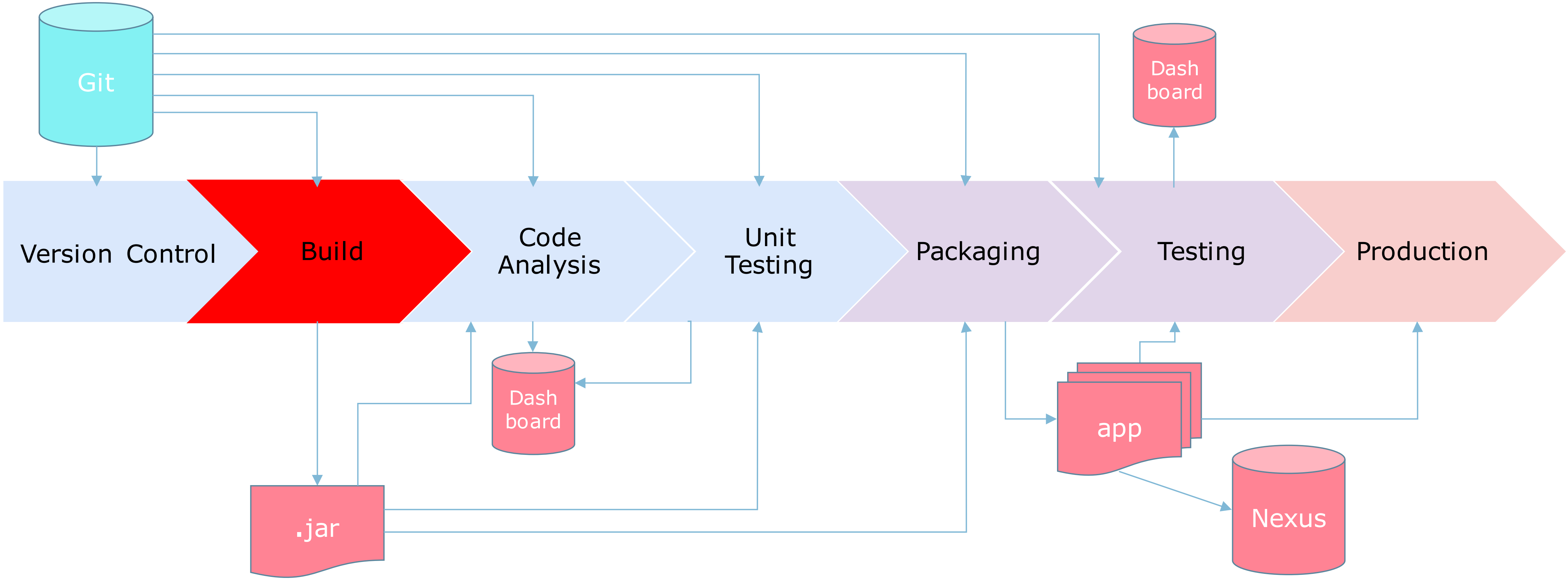


Interacting with the Version Control System

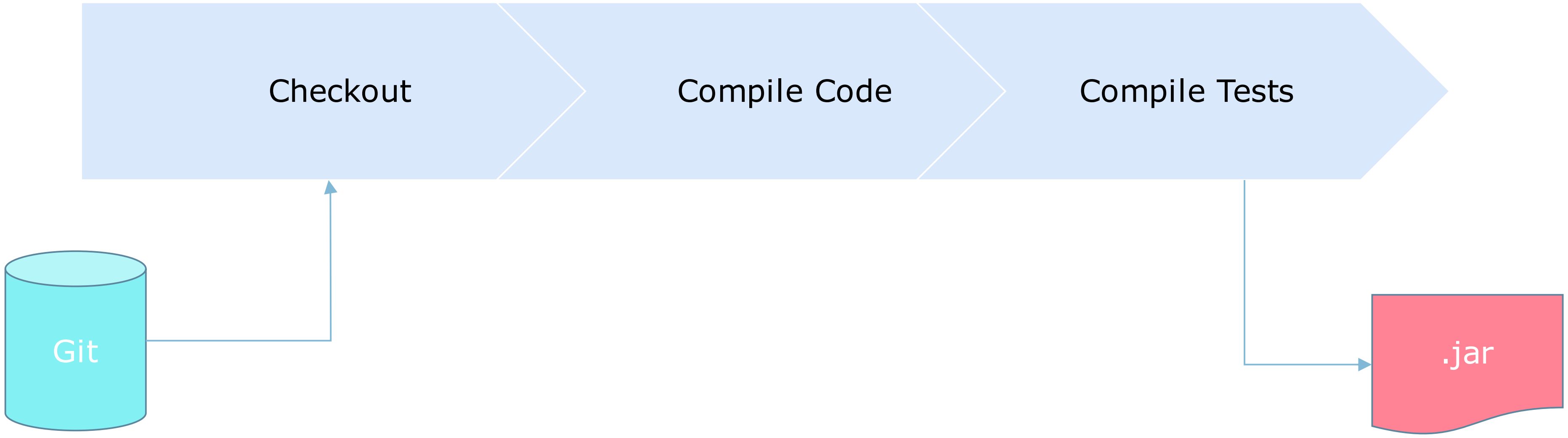


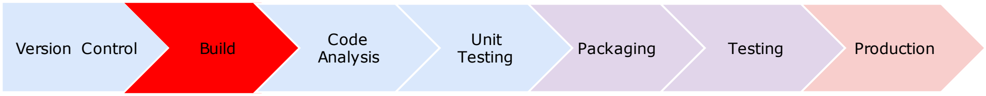


Build – The Automation starts ist work

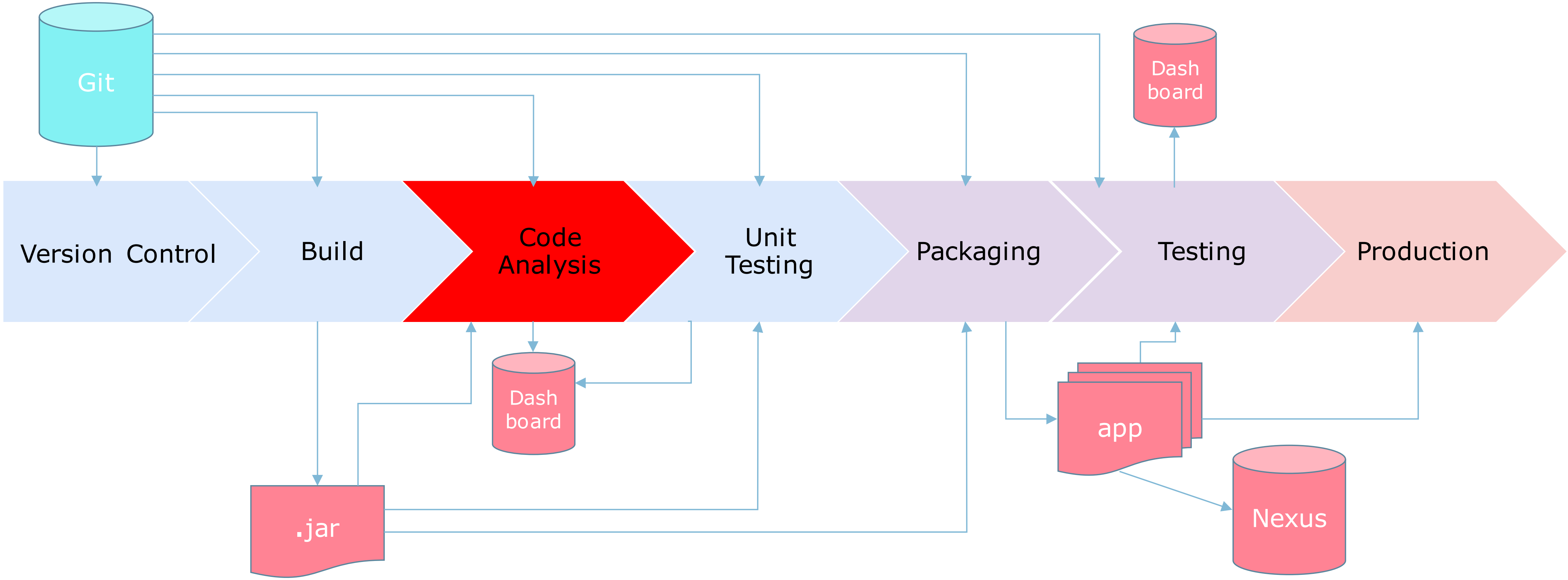


Compiling the Code

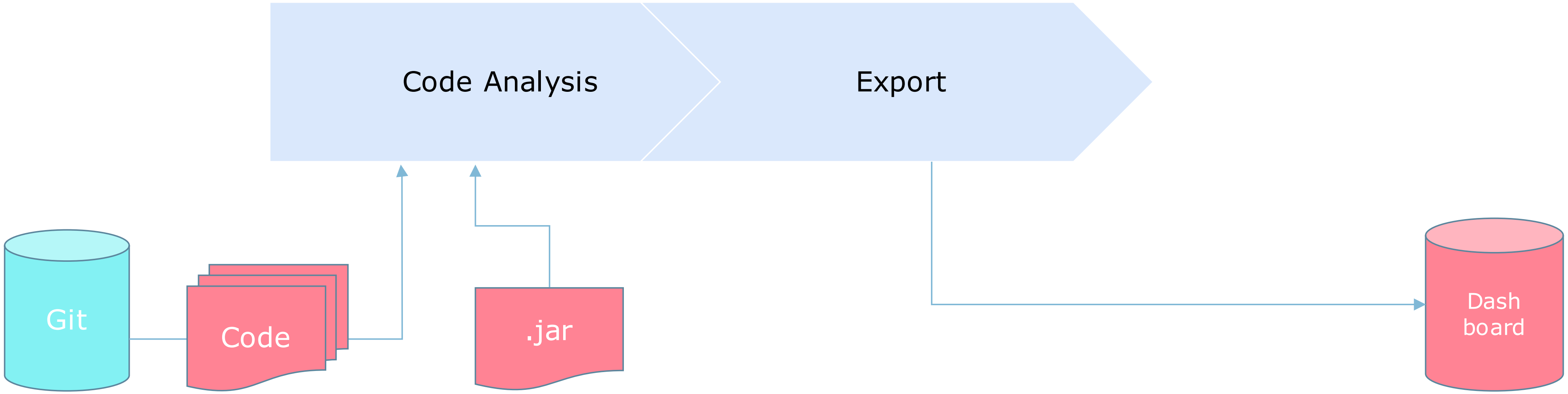


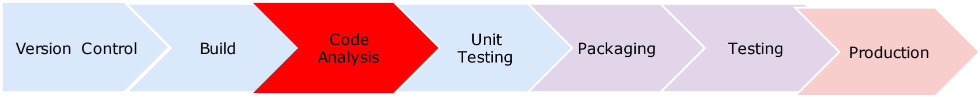


Code Analysis – First Step of Quality Assurance

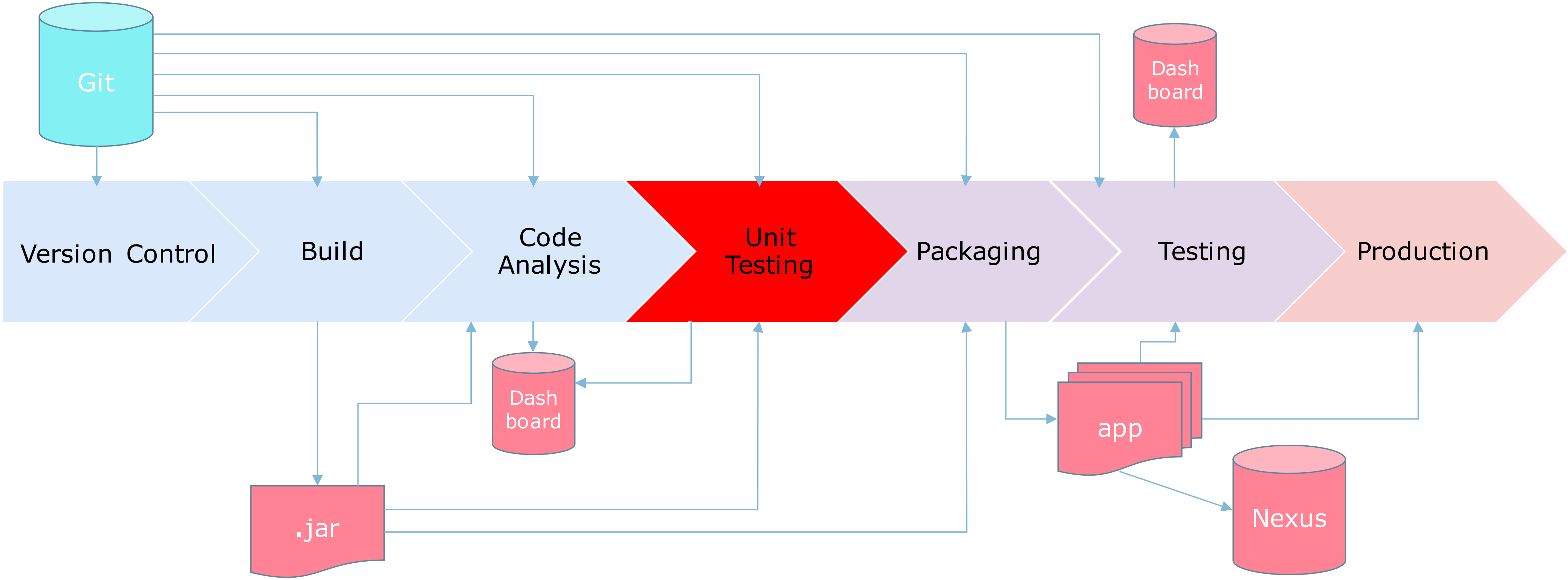


Code Analysis – The Referee Decides if You Followed the Rules

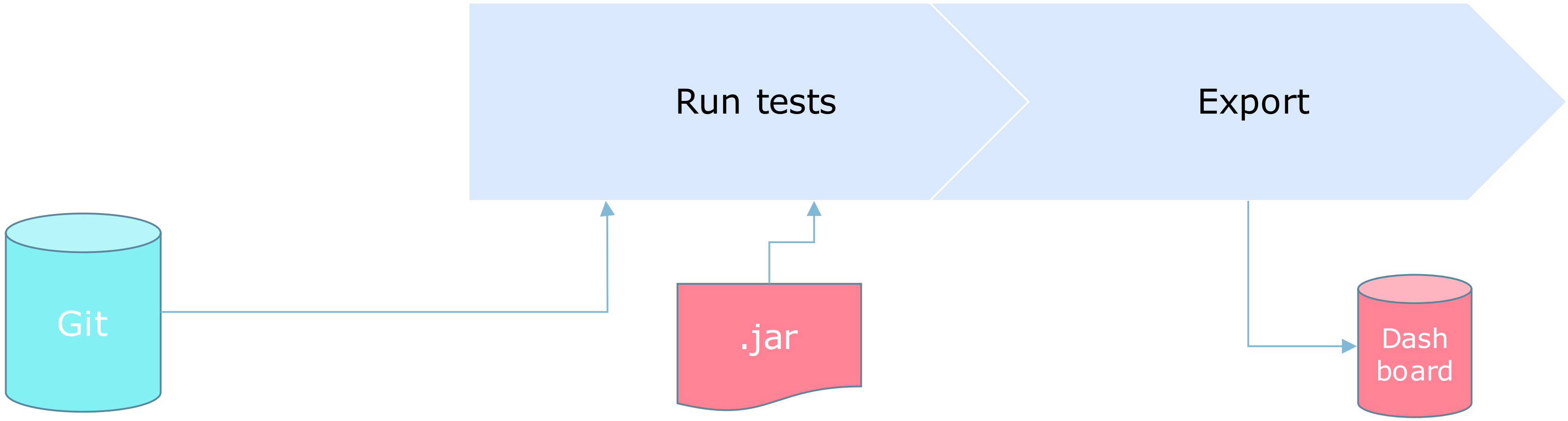


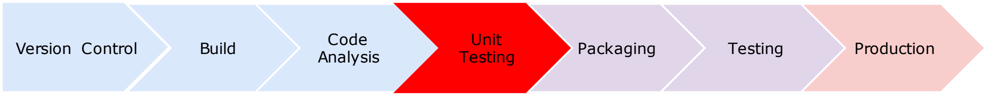


Unit Testing – The Journey of QA goes on

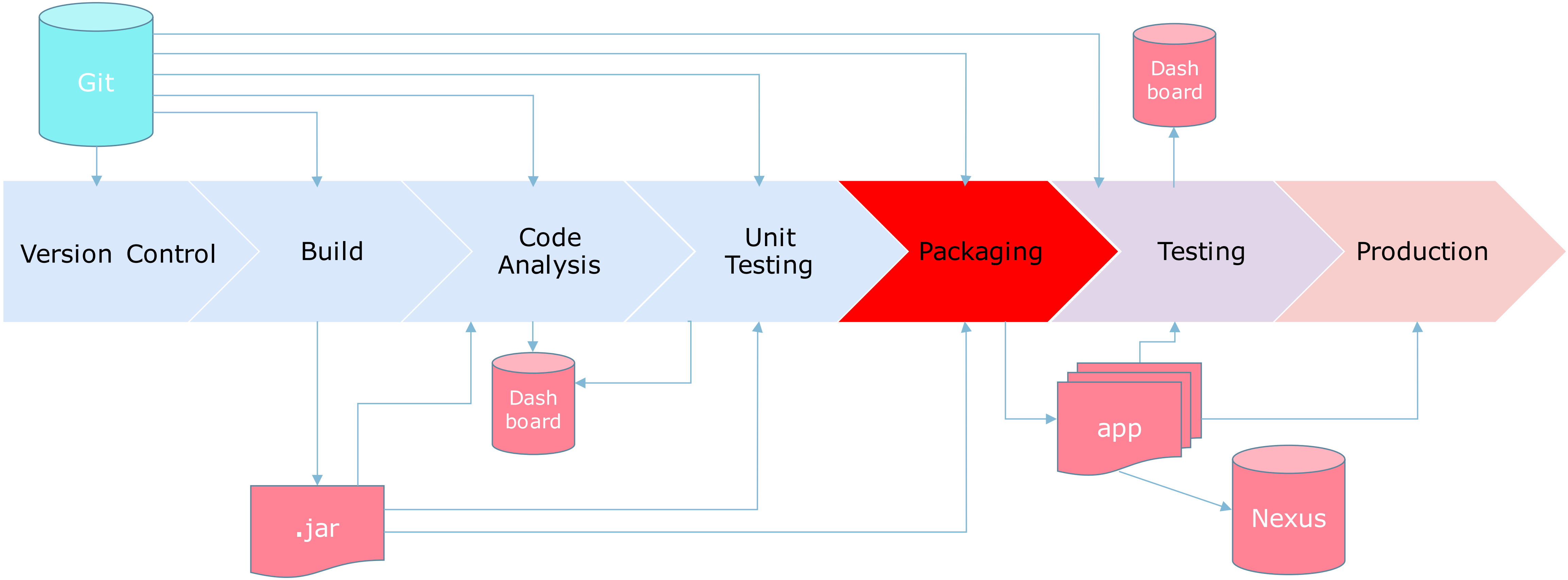


Unit Testing – Do Artefacts Do What They are Intended to Do?

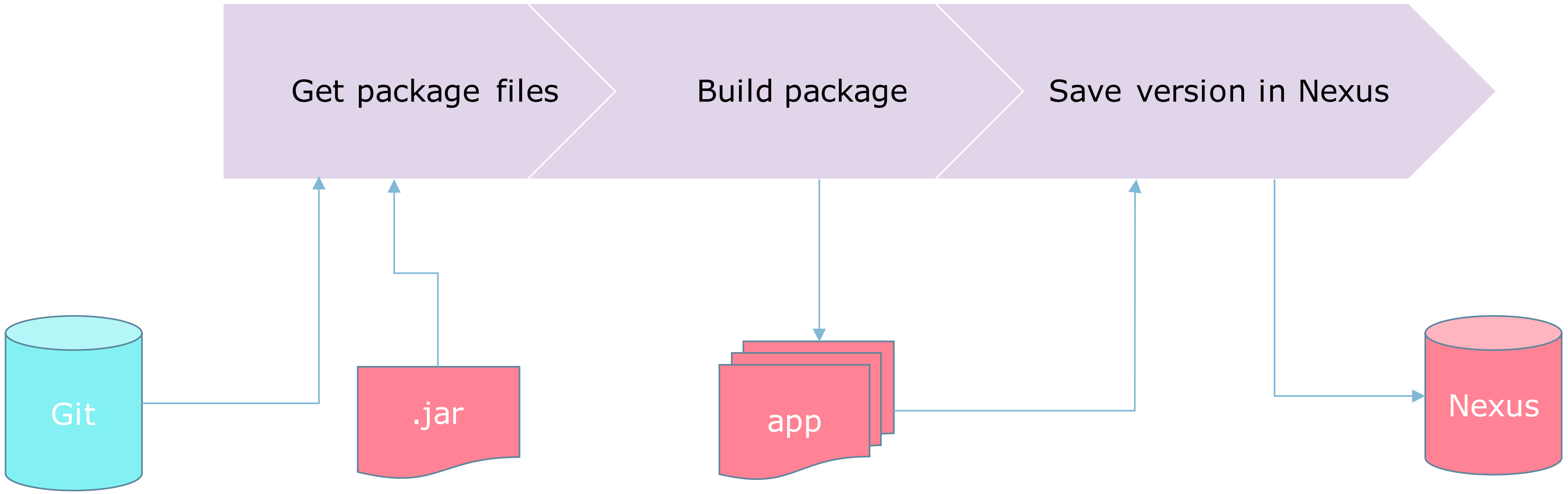




Packaging – Preparation of Deployments



Packaging – Get All Together What is Needed



Get package files

Build package

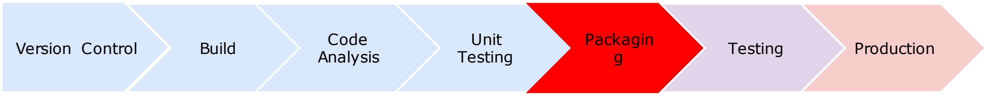
Save version in Nexus

Git

.jar

app

Nexus



Version Control

Build

Code Analysis

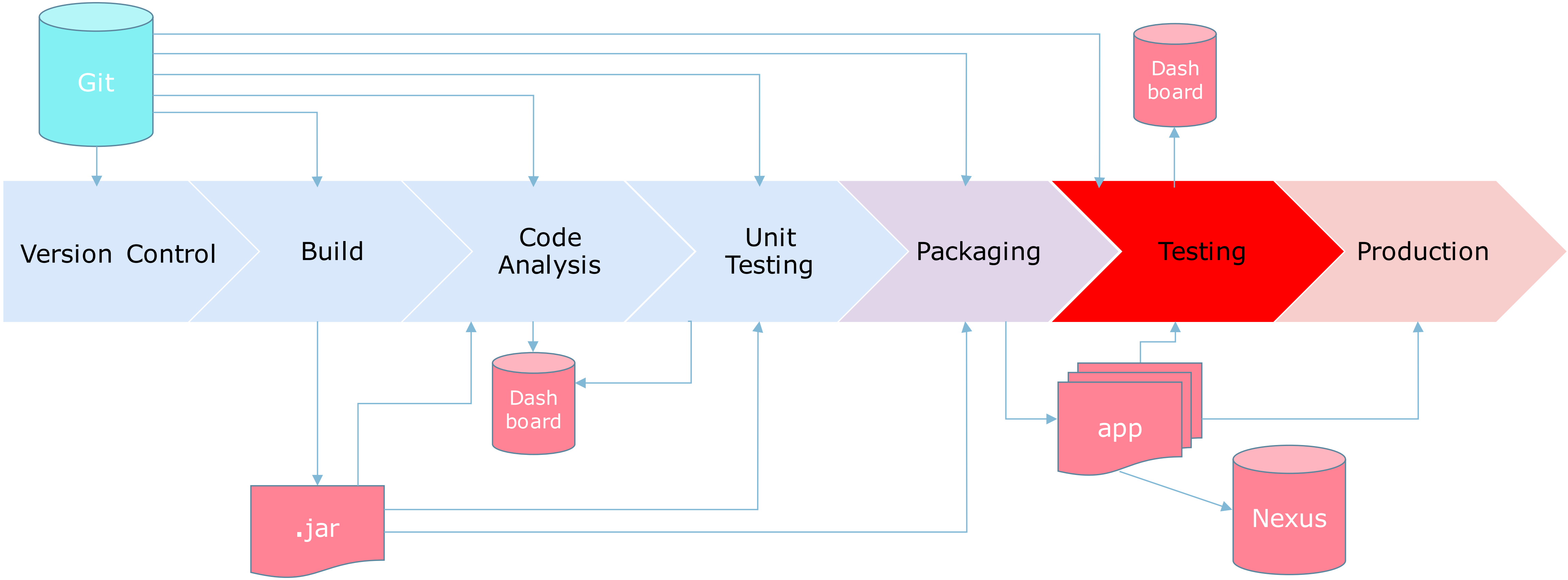
Unit Testing

Packagin g

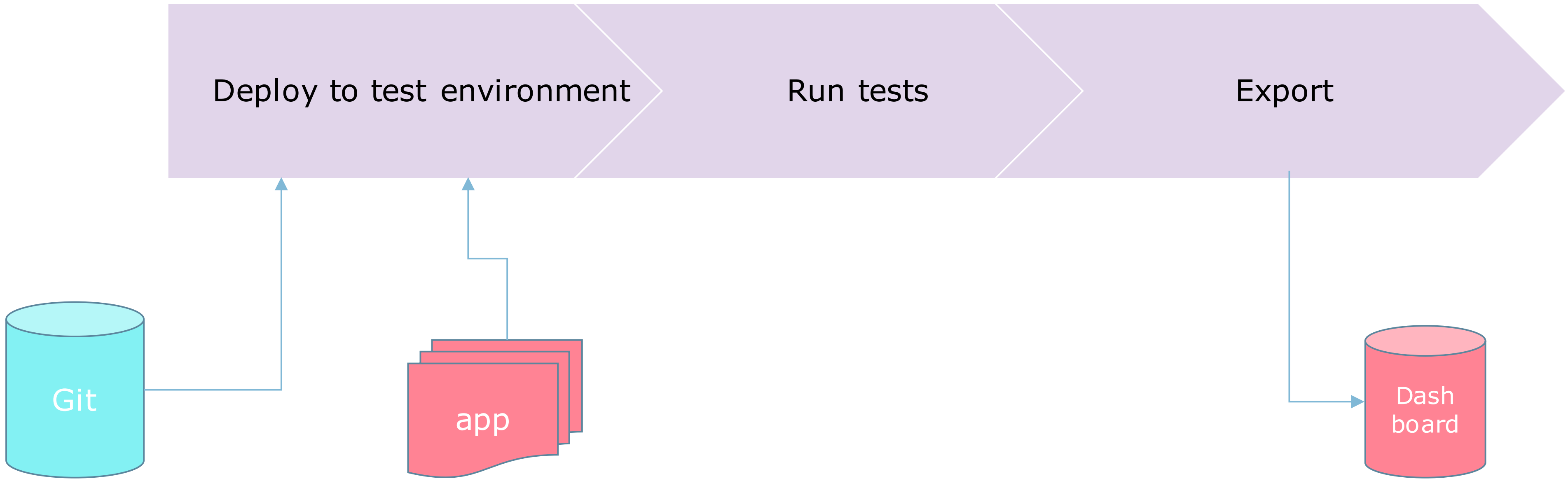
Testing

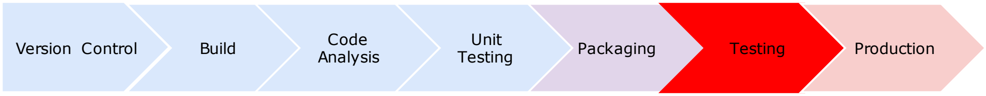
Production

Testing – Does the Software behave as expected

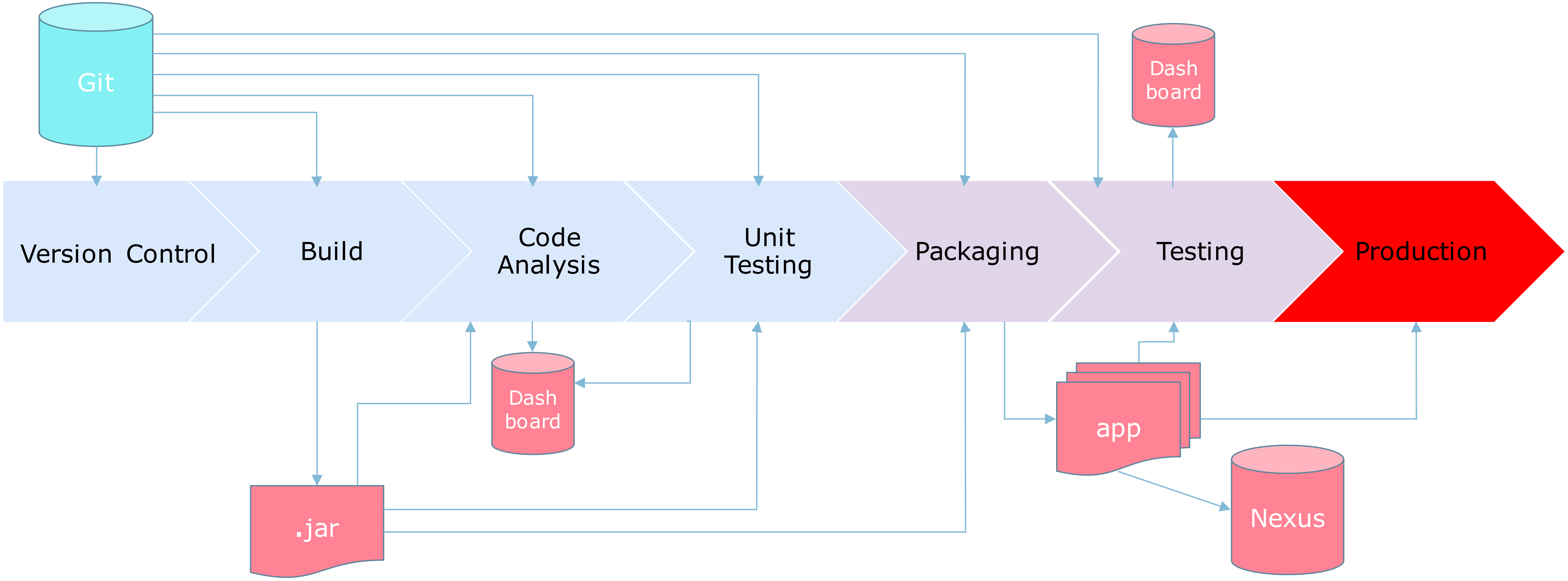


Testing – Let the App Run on Device and Check Functionality

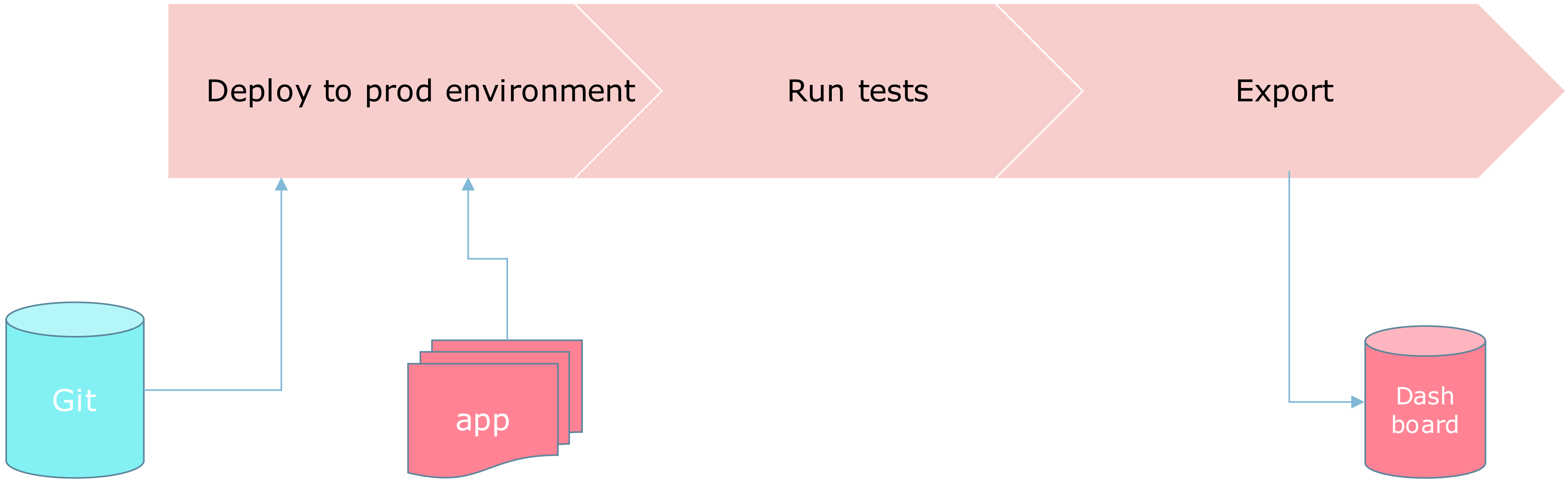


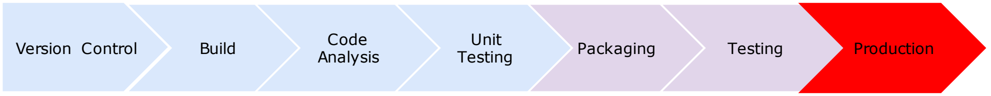


Production – Let`s GoLive



Production – Congratulations, You Went Live





About Capgemini

A global leader in consulting, technology services and digital transformation, Capgemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, Capgemini enables organizations to realize their business ambitions through an array of services from strategy to operations. Capgemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of 200,000 team members in over 40 countries. The Group reported 2016 global revenues of EUR 12.5 billion.

Learn more about us at

www.capgemini.com

People matter, results count.

This message contains information that may be privileged or confidential and is the property of the Capgemini Group.