

## **Spree: Test Strategy**

### **1. Introduction:**

**Spree is an E-commerce (electronic commerce)** application. This application involves the activity of electronically buying products on online services or over the Internet.

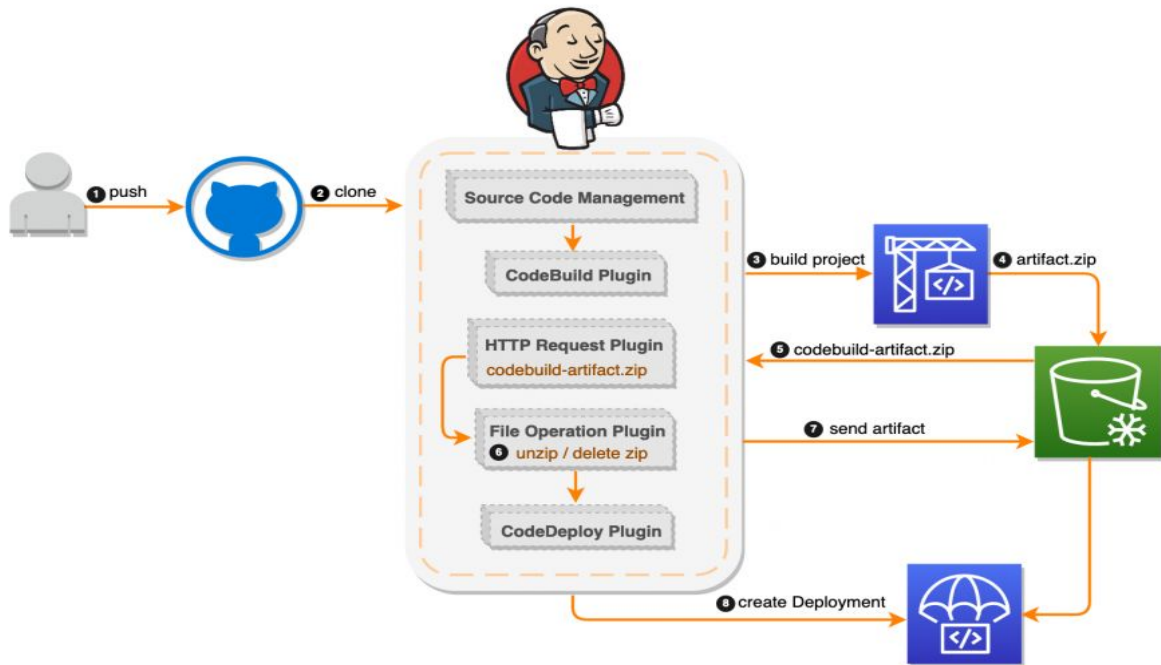
### **2. Featured to be tested:**

- Easy login & registration
  - User-friendly product filtering and sorting
  - Detailed product descriptions
  - Product gallery
  - Shopping cart
  - Shipping options
  - Order summary
  - Secure and easy in-app payments
  - Ordering an invoice and adding company data

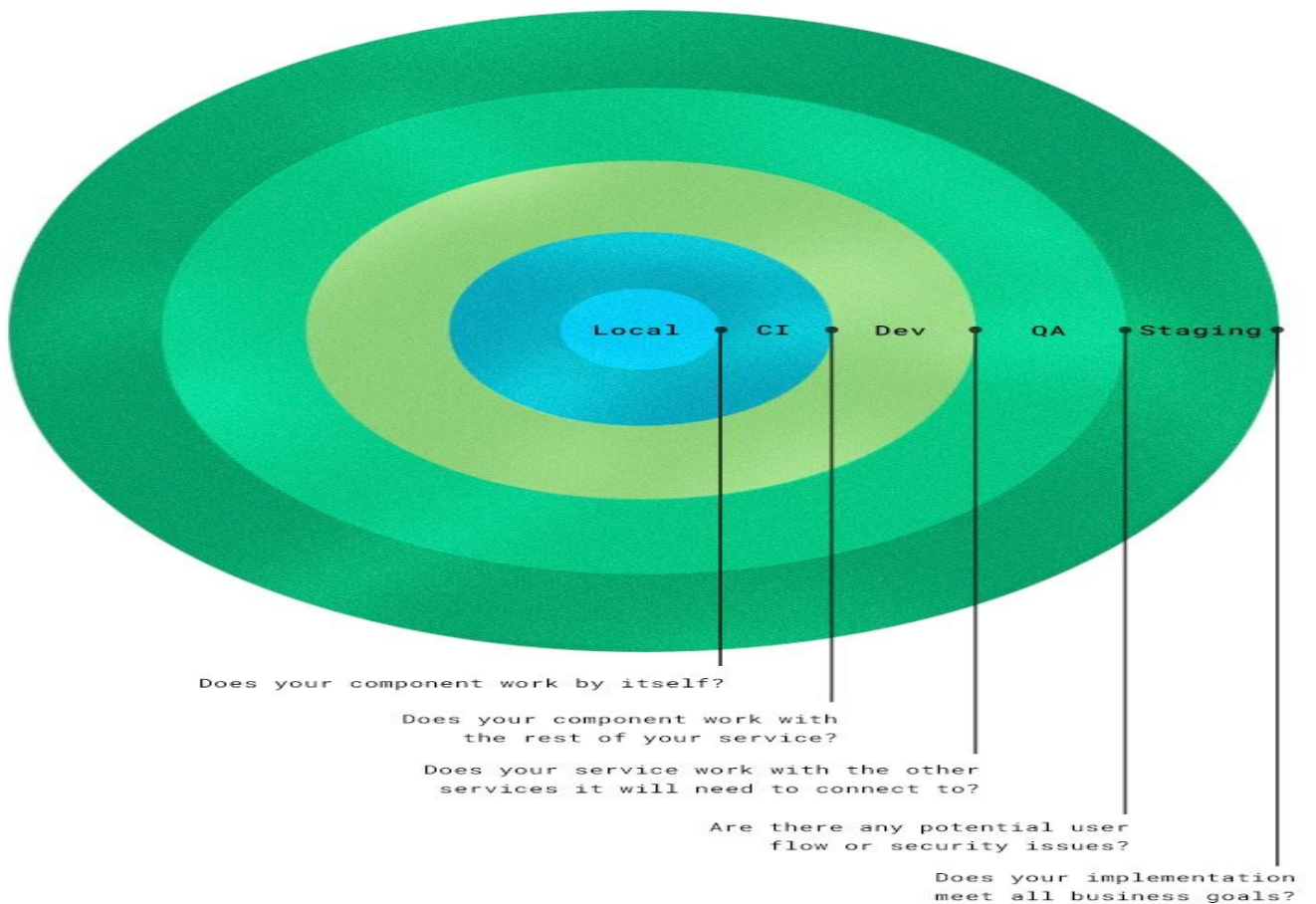
### **3. Featured not in scope of current release:**

- Returns and checking the return status
- Checking the shipping status
- The number of items users are seeing/left on the list
- Checking item availability in specific locations
- Item availability reminder
- Wishlist

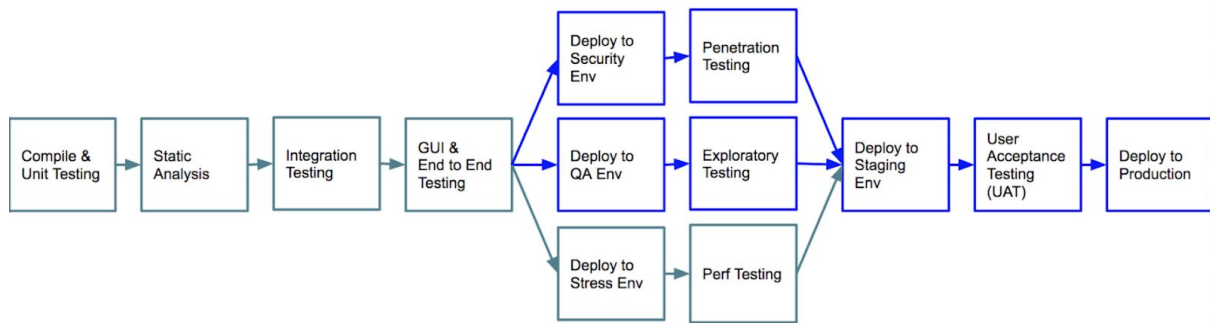
## 4. Build Path to production



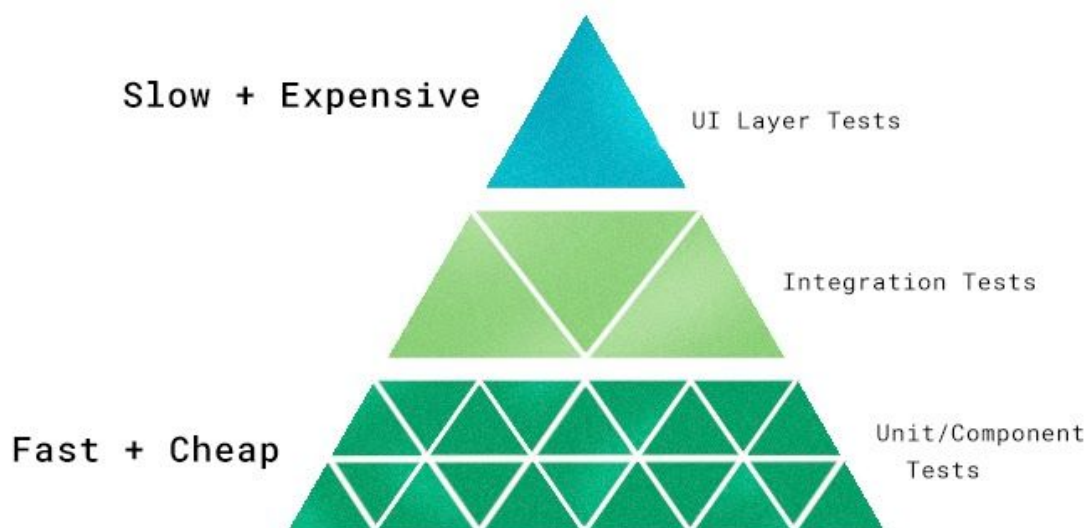
How tests fit into a software development pipeline.



## 5. Build Tasks



## 6. Types of tests and Test Pyramid:



- **Unit/component test:** These cover the smallest possible components, units, or functionalities. They're the cheapest and fastest tests to run since they don't require a lot of dependencies or mocking. These should be done early to get them out of the way.
- **Integration test:** These check how well each unit from the previous stage works with the other components, units, and functionalities. In a broader sense, it can test how services (such as APIs) integrate with one another.

- **UI layer testing:** This is automated browser-based testing which tests basic user flow. It is expensive to set up and slow to run, so it should happen later in the pipeline.
- **Performance testing-** a top priority in Spree application.

## 7. What are we not covering in tests:

Security Testing  
Infrastructure Testing

## 8. Test Tools

UI Automation: Selenium Webdriver, TestNG

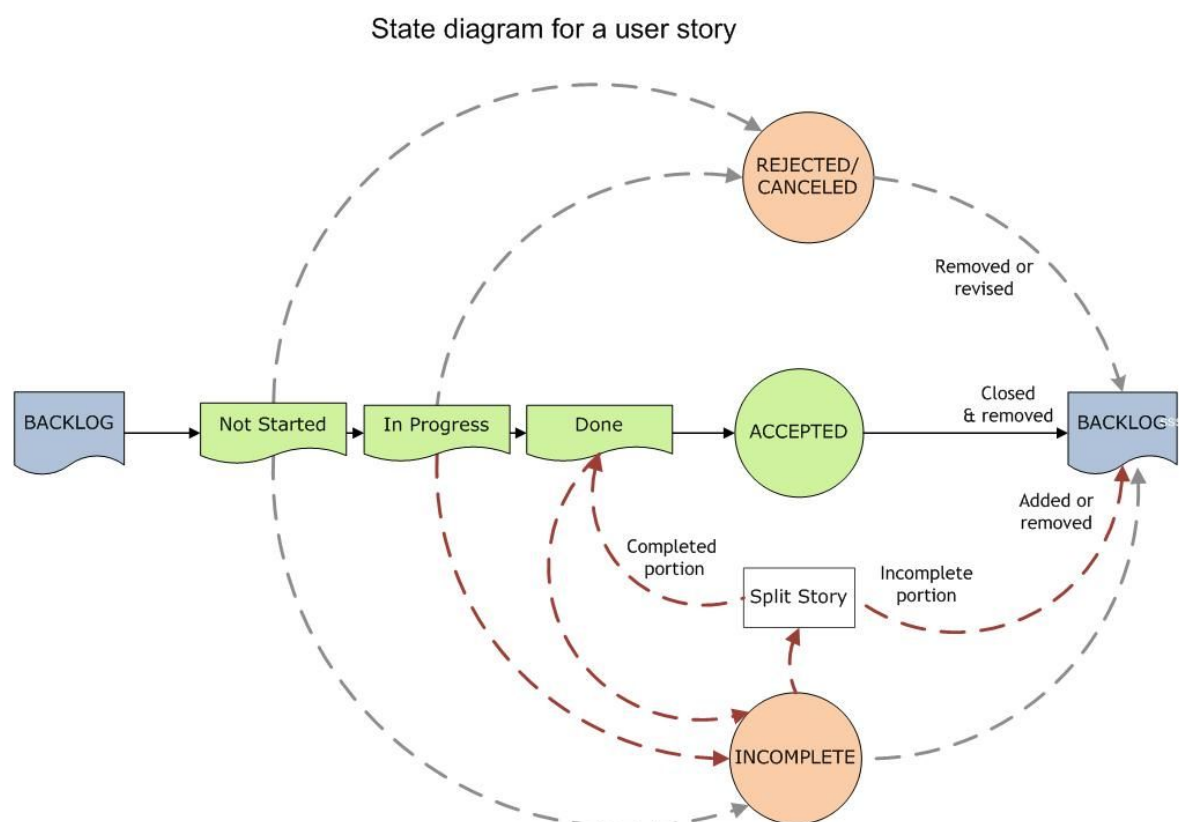
CI: Jenkins

Configuration management and Version Control: GIT & GitHub

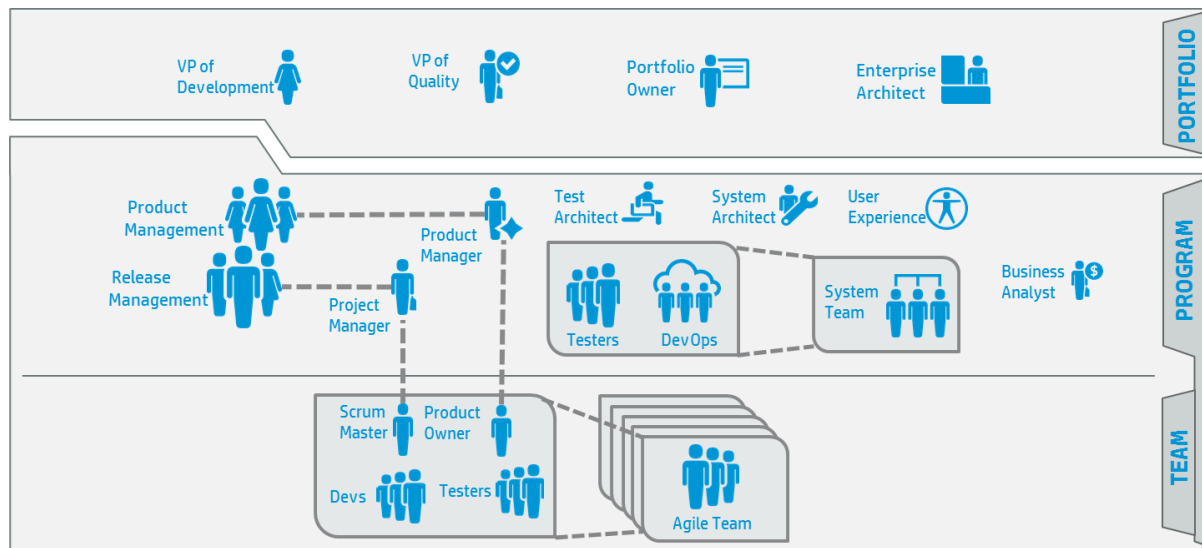
Performance Testing: JMeter

Test Management and Defect Tracking: QC

## 9. Story Lifecycle:



## 10. Architecture level Strategy



## 11. Defect life cycle

**BUG LIFE CYCLE Diagram:**

