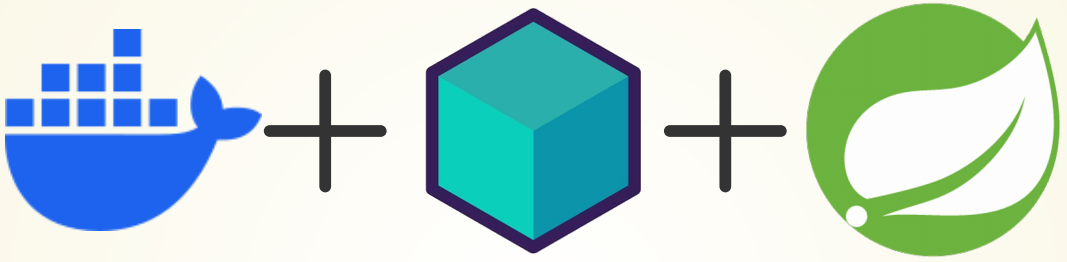


REPRODUCIBLE DEVELOPMENT AND TESTING ENVIRONMENTS WITH SPRING BOOT 3.4

EMÍLIO SIMÕES

Present Technologies

SOFTWARE DEVELOPMENT

SOFTWARE DEVELOPMENT

Software development is:

- Software development is the process of designing, creating, testing, and maintaining computer programs and applications to meet specific user needs or solve problems

DEVELOPMENT ENVIRONMENT

A development environment is:

- A development environment is a workspace equipped with the necessary tools, configurations, and resources for software developers to write, test, debug, and refine code before it's deployed to production

DEVELOPMENT DEPENDENCIES

Applications usually do not live in isolation.

- They require external components like databases, distributed caches, queues, brokers, etc.
- Development cannot be done without those dependencies have been satisfied.

DEVELOPMENT DEPENDENCIES

Option 1: Using mocks

- Using mocks does not reproduce a real production environment
- They are pieces of code that returns some hard coded responses that do not reproduce the actual external component behavior
- It's additional code that needs to be maintained
- It's only a temporary solution because the app will eventually have to be connected to the dependency

DEVELOPMENT DEPENDENCIES

Option 2: Use a running service instance

- External dependencies need to be running either locally or as a remote service for the app connect to them and to run successfully
- The external component instances need to be started and stopped manually

DEVELOPMENT DEPENDENCIES

Option 3: Use embedded instances

- Embedded and in-memory databases are a more lightweight approach which are more easier to automate and maintain
- Different database engines have different features and behaviors which can cause the app not to behave like in the production environment
- Some features are not supported in embedded and in-memory databases
- Other external components might not have an embedded implementation

DEVELOPMENT DEPENDENCIES

Option 4: Use docker containers

- Having a docker compose file for the development solves most of the problems from the other options
- We can use the exact same external resources and versions as in the production environment
- But… it needs to be started, cleaned and stopped manually

SPRING BOOT DOCKER COMPOSE SUPPORT

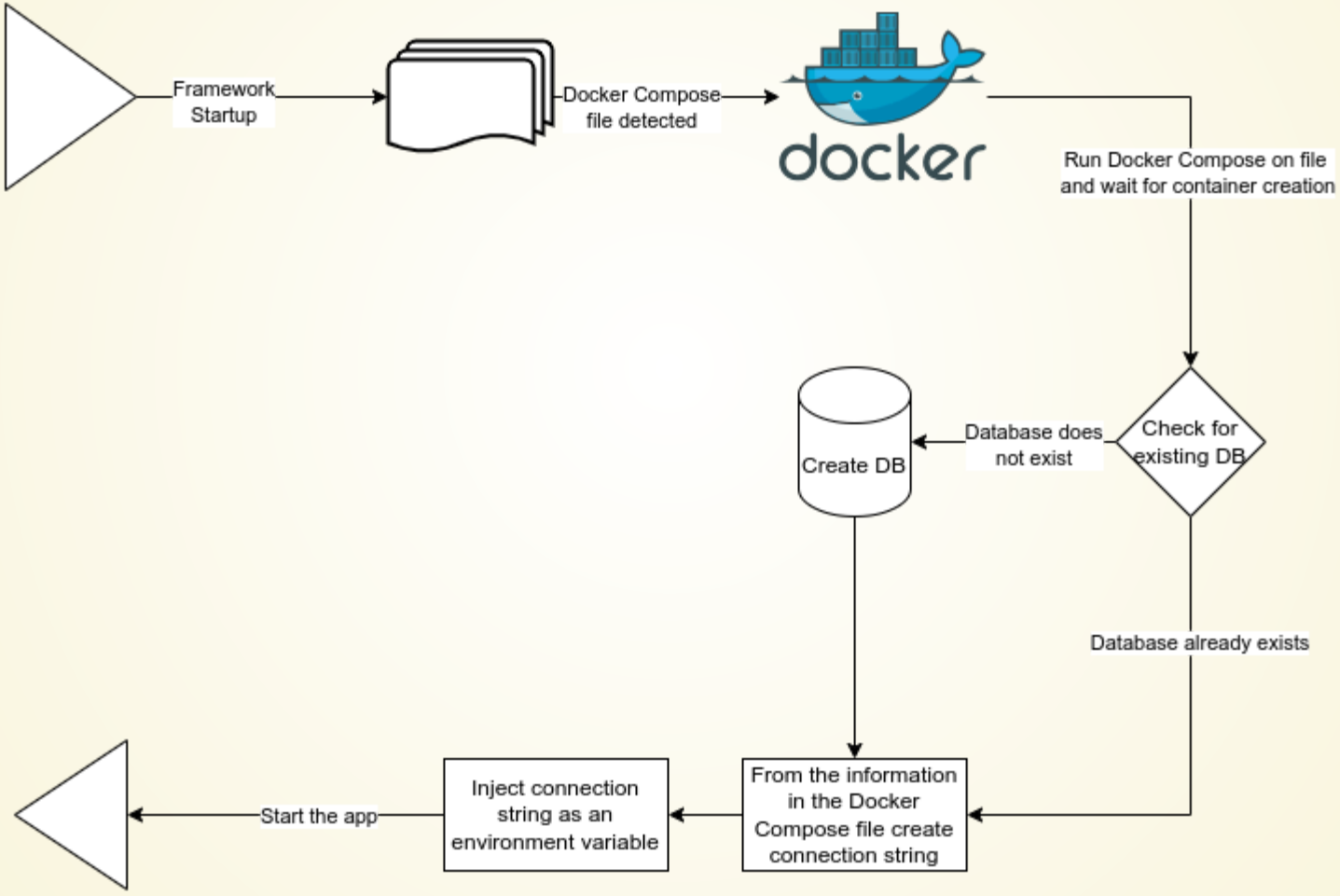
- Spring Boot Docker compose support it's a set of libraries that can be included in the Spring Boot project
- It detects any docker compose file in the project root directory and automatically launches the containers
- Can be used with any application or service that can be containerized

DEMO TIME

WHAT HAPPENED HERE

- Spring detected a docker compose file with a postgres database configuration
- It run docker compose on that file creating the containers
- Created a database if none existed
- From the information in the docker compose created a connection string and injected it in the application configuration
- From there on the application just behaved like a normal spring application

WHAT HAPPENED HERE



SOFTWARE TESTING

SOFTWARE TESTING

Software testing is:

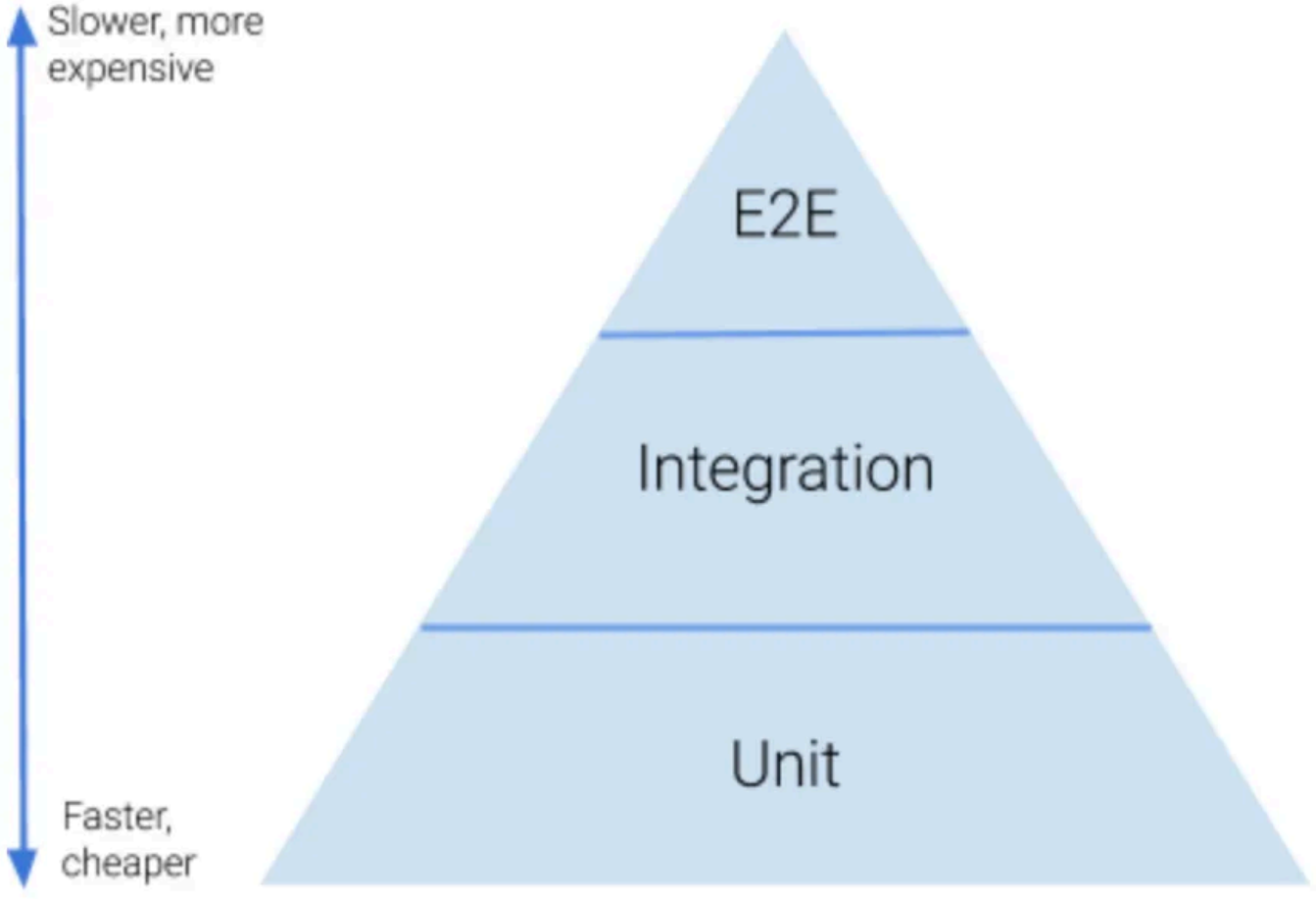
- The process of checking for errors in an application
- Verifying if the application behaviour matches the desired expectations

SOFTWARE TESTING

Software testing is important to:

- Identify defects early
- Improve product quality
- Contribute to product security
- Make it easier to add or change features
- Increase the customer satisfaction

TYPES OF SOFTWARE TESTING



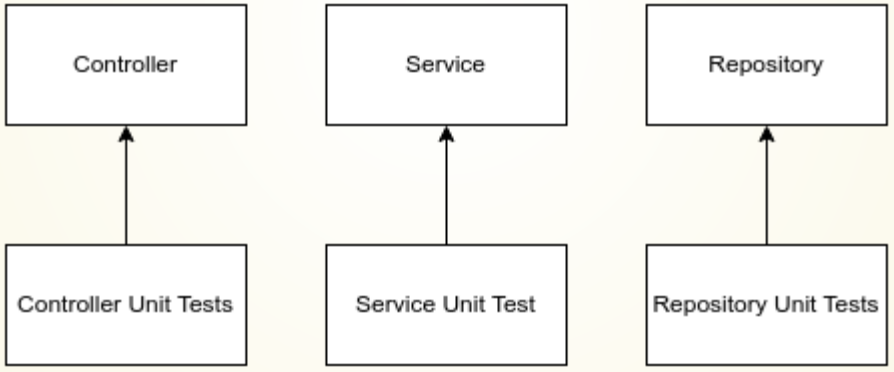
TYPES OF SOFTWARE TESTING

Unit tests

- Test individual components (code units) in isolation to check if they work as expected
- Are fast to execute
- Are executed very often during the development process

TYPES OF SOFTWARE TESTING

Unit tests



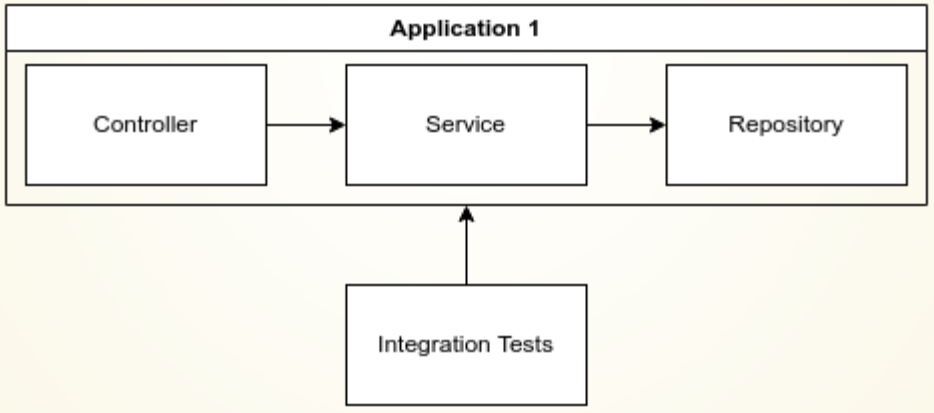
TYPES OF SOFTWARE TESTING

Integration tests

- Test the interactions between the several application components or external components and ensures that it is working as expected
- Are slower to execute
- Run fewer times (e.g. push, pipeline, etc.)

TYPES OF SOFTWARE TESTING

Integration tests



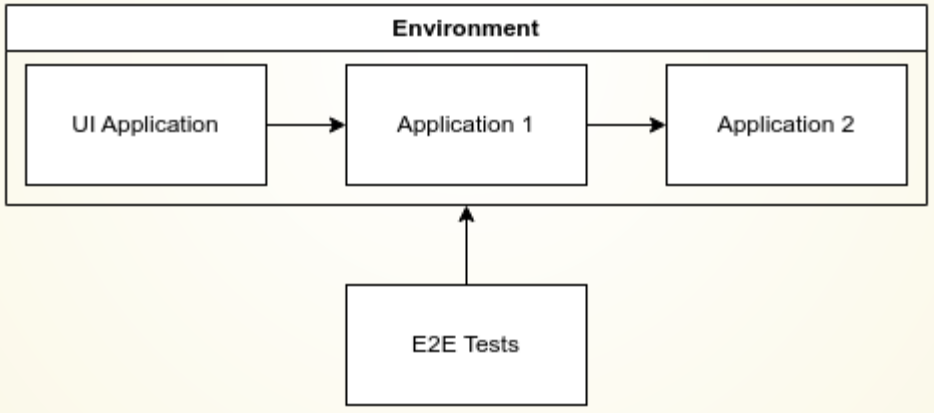
TYPES OF SOFTWARE TESTING

E2E tests

- Tests the entire application or applications from start to finish, usually in a test environment to simulate a real-world usage
- Usually take a lot of time to exectute
- Are run in specific points of the software lifecycle or as a daily job

TYPES OF SOFTWARE TESTING

E2E tests

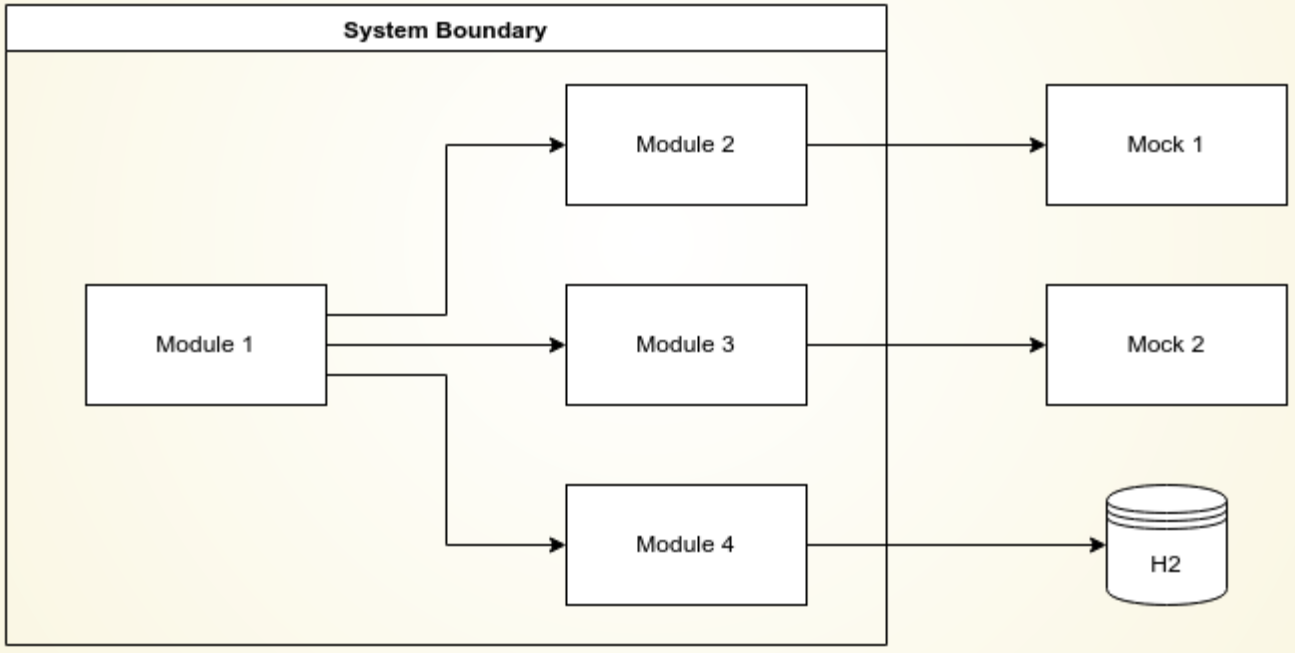


EXTERNAL COMPONENTS PROBLEM

Applications usually do not live in isolation.

- They require external components like databases, distributed caches, queues, brokers, etc.
- Testing an application that requires one or more of these external components introduces some challenges.

EXTERNAL COMPONENTS PROBLEM



EXTERNAL COMPONENTS PROBLEM

Option 1: Using mocks and fakes

- Using mocks or fakes does not reproduce a real production environment
- They are pieces of code that returns some hard coded responses that do not reproduce the actual external component behavior
- It's additional code that needs to be maintained

EXTERNAL COMPONENTS PROBLEM

Option 2: Use a running service instance

- External components need to be running either locally or as a remote service for the tests to run successfully
- Proper setup scripts need to be maintained for the tests only and can be hard to get a clean environment for each test run (e.g. clean database)
- The external component instances need to be started and stopped manually

EXTERNAL COMPONENTS PROBLEM

Option 3: Use embedded instances

- Embedded and in-memory databases are a more lightweight approach which are more easier to automate and maintain
- Different database engines have different features and behaviors which can cause the tests not to be conclusive
- Some features are not supported in embedded and in-memory databases
- Other external components might not have an embedded implementation

EXTERNAL COMPONENTS PROBLEM

Option 4: Use docker containers

- Having a docker compose file for the tests solves most of the problems from the other options
- We can use the exact same external resources and versions as in the production environment
- But… it needs to be started, cleaned and stopped manually

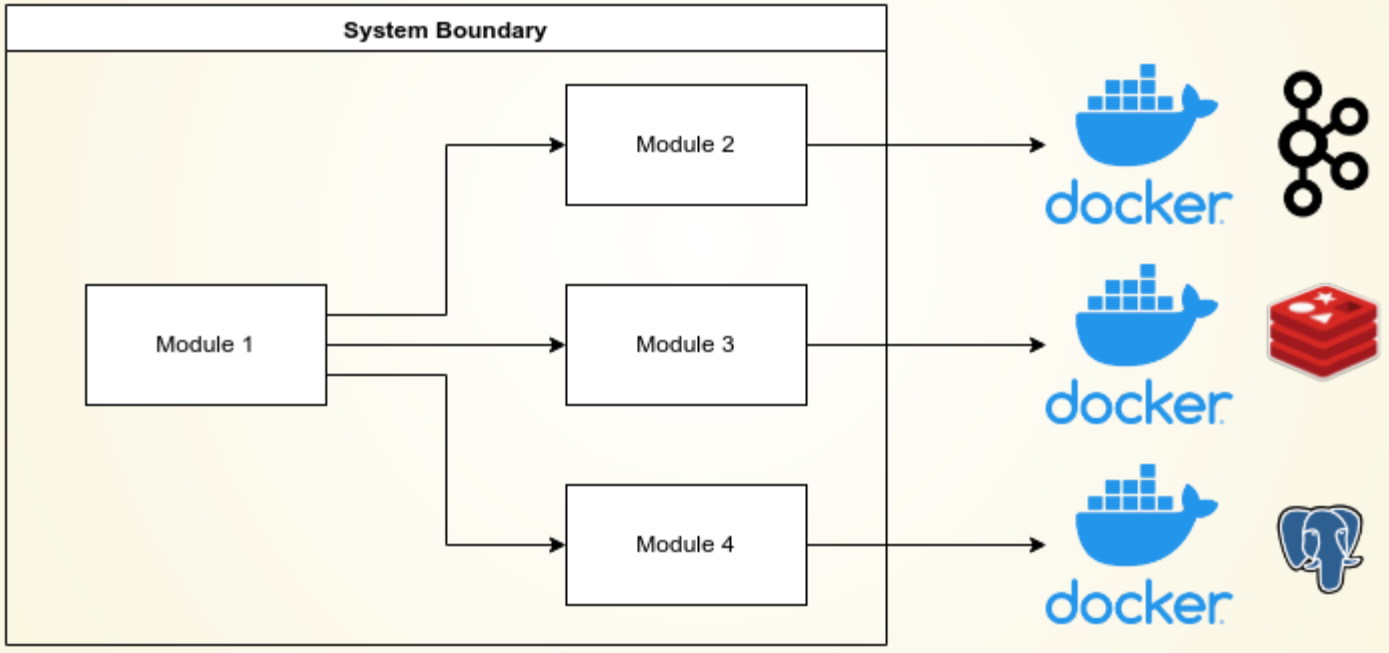
EXTERNAL COMPONENTS PROBLEM

- Spring Boot Docker compose support is not a solution here
- It's not design to work with test, it's designed to create a reproducible development environment
- It does handle issues like cleaning up after the test, this is only a test requirement
- While developing you might even want to persist data between sessions

TESTCONTAINERS TO THE RESCUE

- Testcontainers is a library that can start and stop one or more docker containers with the same resource as we use in our production environment
- It supports several languages like Java, Go, .NET, Node.js, Python and Rust
- Is framework agnostic so it can be used anywhere
- Can be used with any application or service that can be containerized

TESTCONTAINERS TO THE RESCUE



DEMO TIME

QUESTIONS?

PRESENT
TECHNOLOGIES

THANKS FOR PARTICIPATING