ACADEMIND

# What Is Testing?

Verify “if something works as intended”

| Manual Testing | Automated Testing |
|---|---|
| Tedious & cumbersome | Initial effort (write tests), no effort thereafter |
| Error-prone | Predictable & consistent |
| Often incomplete (not all scenarios covered) | High / complete code & scenario coverage can be achieved |

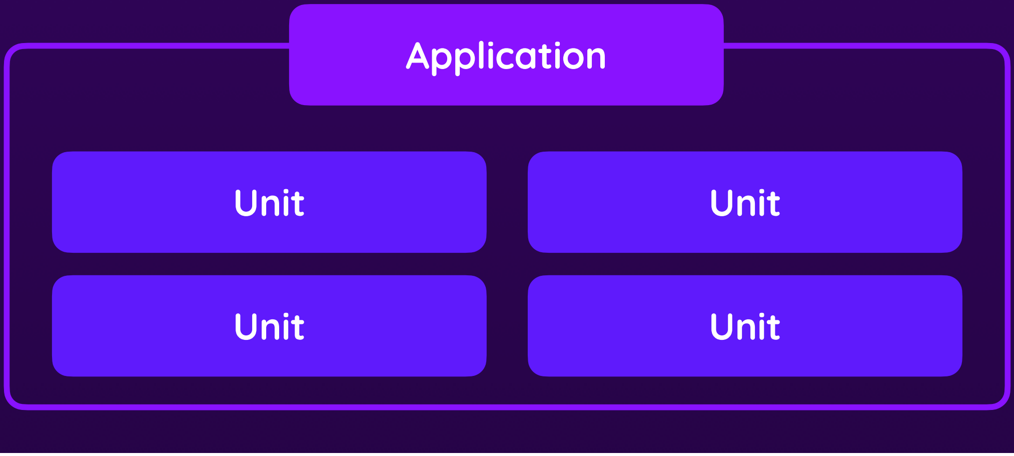
# What Are Unit Tests?

Unit

A **building block** of your app

Ideally, the **smallest possible** building block

e.g., a function, a class, a component, ...



Application

Unit

Unit

Unit

Unit

App = combination of all units

If all units were tested, the overall app should work → Backed up by **integration tests**

Changes are always tested against all units to avoid bugs

# Why Unit Testing?

Avoids endless amounts of manual testing

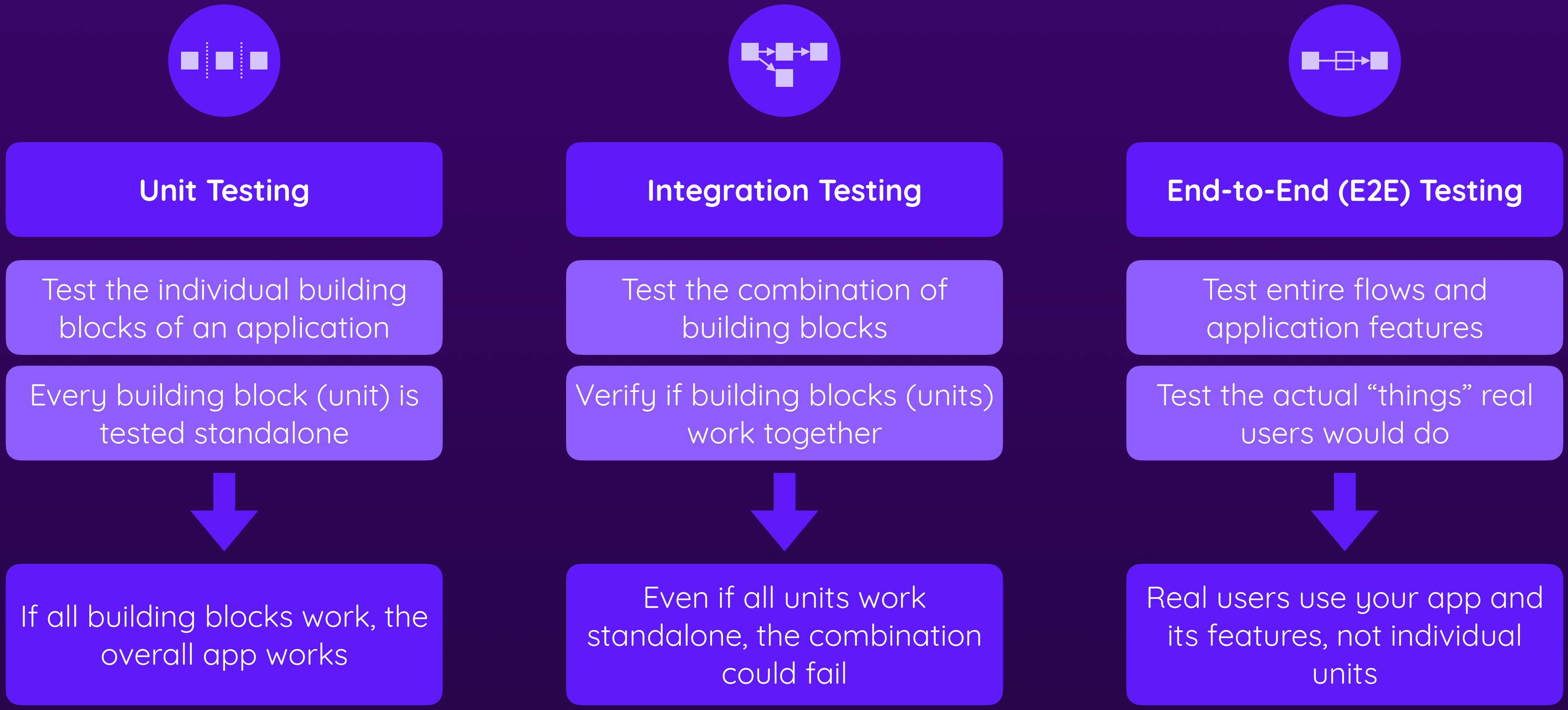
Allows you to cover (close to) 100% of your code & scenarios

Code changes are tested against all scenarios (almost) instantly

Write cleaner & better code (because testing then becomes easier)

ACADEMIND

# Unit, Integration & End-to-End (E2E) Testing



## Unit Testing

Test the individual building blocks of an application

Every building block (unit) is tested standalone

If all building blocks work, the overall app works

## Integration Testing

Test the combination of building blocks

Verify if building blocks (units) work together

Even if all units work standalone, the combination could fail

## End-to-End (E2E) Testing

Test entire flows and application features

Test the actual “things” real users would do

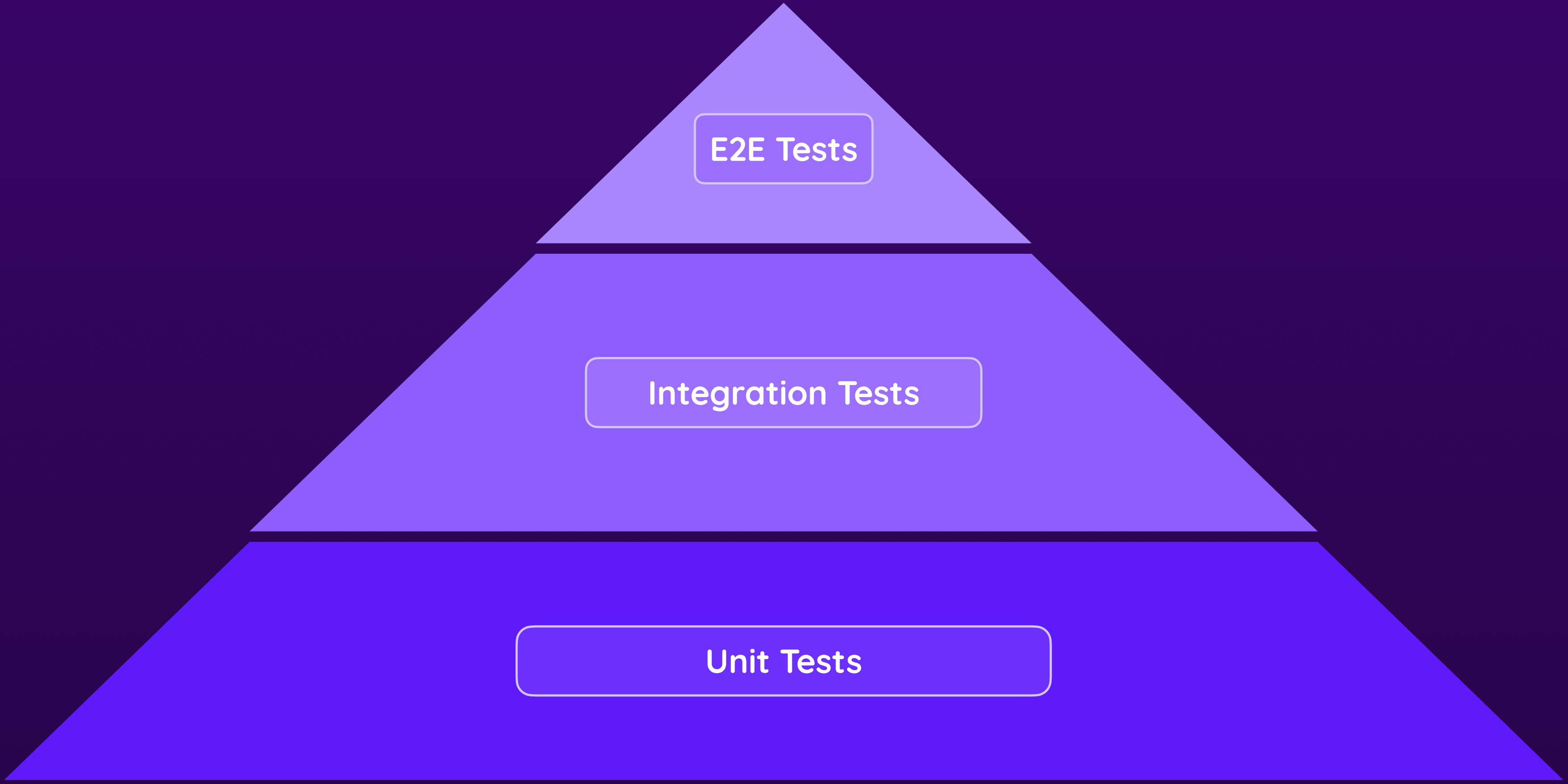
Real users use your app and its features, not individual units

# Unit, Integration & End-to-End (E2E) Testing

You should combine all kinds of tests

| Unit Testing | + | Integration Testing | + | End-to-End (E2E) Testing |
|---|---|---|---|---|
| Quickly spot and pinpoint breaking changes & errors | | Test (parts of) processes & combinations of units | | Test realistic user flows & behaviors |
| Ignore actual user flows and interferences | | Spotting the exact root of an error can be tricky | | Covering all possible behaviors can be challenging |

# The Testing Pyramid



# About This Course

- Getting Started
- Project Setup & Required Tools
- Testing Fundamentals
- Writing Good Tests
- Advanced Testing Concepts
- Mocks & Spies
- Diving Deeper into Mocks & Spies
- Tests & The DOM

# How To Get The Most Out Of The Course

## Watch the videos

At your own pace

Speed up, slow down, pause & rewind!

## Code along

Pause & practice

Experiment & build your own demo apps

## Use the course resources & attachments

See next lecture!

Use the attached code to compare & replace

## Help each other in the Q&A section

Ask & reply to others

By replying to others, you'll learn way more!