

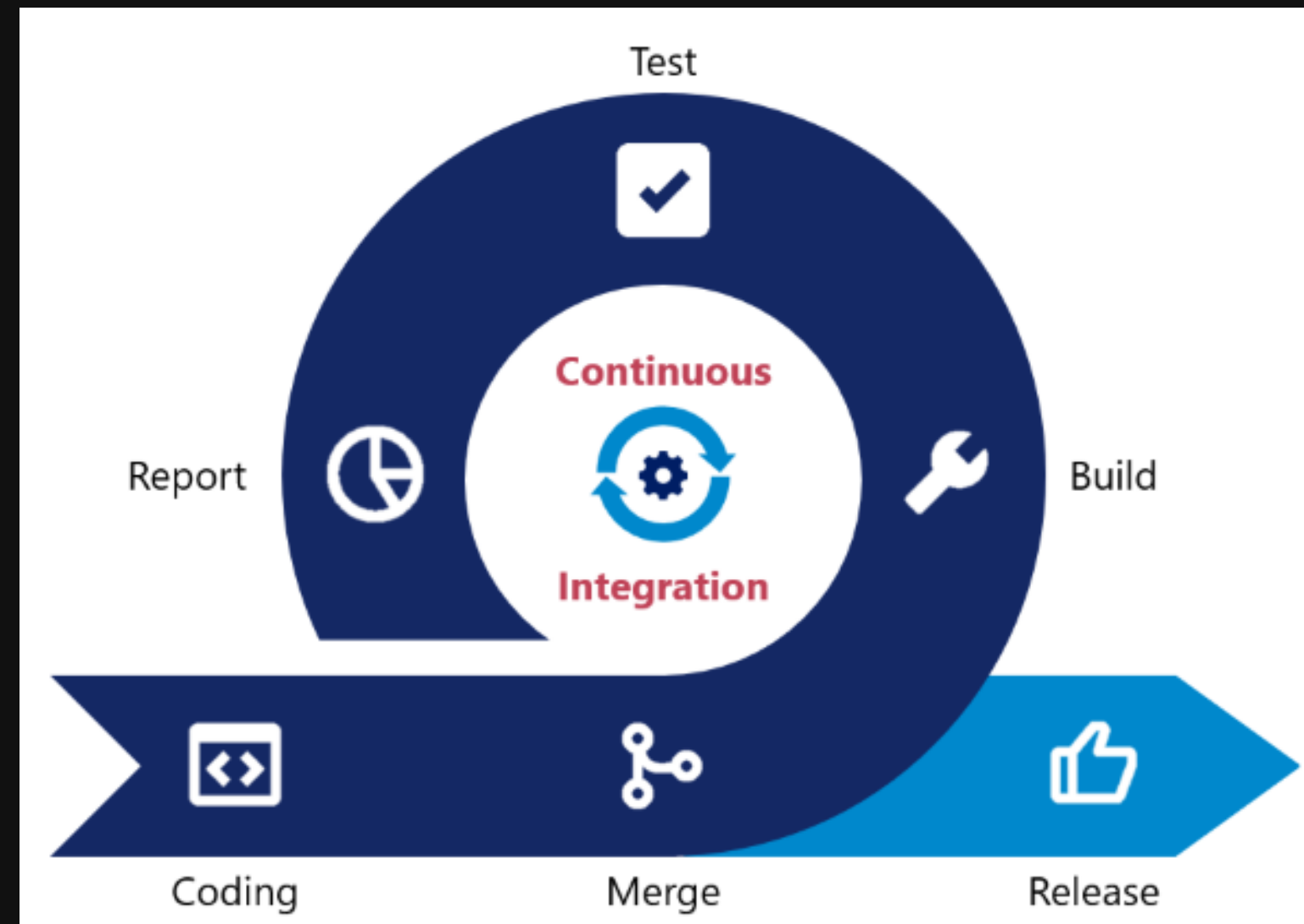
COMP1531

5.1 - SDLC Testing - Continuous Integration

Continuous Integration

Continuous integration: Practice of automating the integration of code changes from multiple contributors into a single software project.

Continuous Integration



Continuous Integration

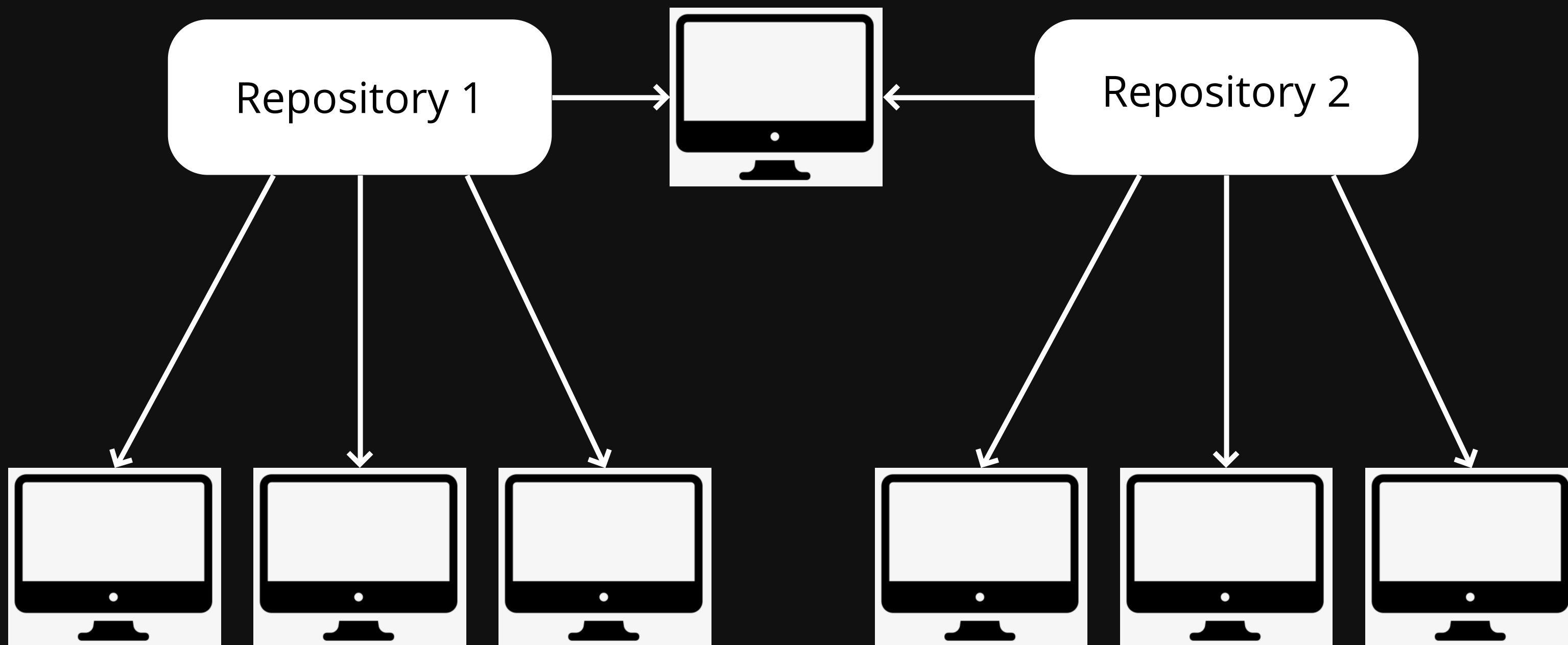
Key principles and processes:

1. Write tests:
 1. Ideally tests for each story
 2. Broad tests: unit, integration, acceptance, UI tests
2. Use code coverage checkers
3. Merge and integrate code as often as possible
4. Ensure the build always works (i.e. is "green")

How it works

- Typically tests will be run by a "runner", which is an application that works with your version control software (git) to execute the tests. This is because tests can require quite resource intensive activities
 - Gitlab: No runners built in
 - Bitbucket: Runners built in

Broad Architecture



CI: Readings

We will assume you have read the following items:

- <https://about.gitlab.com/product/continuous-integration/>
- <https://www.atlassian.com/continuous-delivery/continuous-integration/how-to-get-to-continuous-integration>

Continuous integration, gitlab

Gitlab, like many source control tools, has a way of doing continuous integration. An [overview is here](#) and a [start guide is here](#).

There is quite a lot of variance and depth to this, so we will not cover it in any detail besides high level

A simple example [can be found here](#).

Continuous integration, gitlab

Let's do a demo setting up CI with gitlab.