Podczas oglądania filmiku z automatyzacji procesów budowania, testowania i wdrażania aplikacji w C++, zacząłem rozumieć, jak istotne są te kroki w cyklu życia oprogramowania. Filmik pokazał, jak za pomocą narzędzi takich jak CMake, Google Test i Jenkins można zautomatyzować cały proces

Pierwszym etapem, który omówiono, jest budowanie aplikacji. W przypadku C++ do zarządzania procesem budowania często używa się narzędzia CMake. CMake generuje pliki makefile lub pliki projektu dla różnych IDE, co ułatwia kompilację projektu na różnych platformach.

Przykład prostego pliku CMakeLists.txt:

cmake

cmake_minimum_required(VERSION 3.10)

project(MyApp)

set(CMAKE_CXX_STANDARD 17)

Dodajemy główny plik źródłowy

add_executable(MyApp main.cpp)

Ten plik konfiguruje CMake do stworzenia projektu C++ z plikiem źródłowym main.cpp i ustawieniem standardu C++ na wersję 17.

Testowanie aplikacji

Następnie omówiono testowanie aplikacji. Testowanie jednostkowe w C++ można zautomatyzować za pomocą Google Test. Jest to popularna biblioteka do pisania testów jednostkowych.

Przykład prostego testu w Google Test:

Najpierw trzeba zainstalować Google Test i dodać go do projektu. Oto fragment pliku CMakeLists.txt do integracji Google Test:

Dodaj Google Test do projektu

```
include(FetchContent)
FetchContent_Declare(
  googletest
  URL https://github.com/google/googletest/archive/release-1.10.0.zip
)
FetchContent_MakeAvailable(googletest)

enable_testing()

Dodajemy plik testowy
add_executable(MyAppTest test.cpp)
target_link_libraries(MyAppTest gtest_main)

add_test(NAME MyAppTest COMMAND MyAppTest)
```

Następnie, przykładowy test jednostkowy w pliku test.cpp może wyglądać tak:

```
#include <gtest/gtest.h>
#include "calculator.h"

TEST(CalculatorTest, Addition) {
  Calculator calc;
  EXPECT_EQ(calc.add(2, 3), 5);
}

int main(int argc, char **argv) {
  ::testing::InitGoogleTest(&argc, argv);
  return RUN_ALL_TESTS();
}
```

W tym przykładzie testujemy metodę add z klasy Calculator.

Wdrażanie aplikacji

Ostatnim etapem, który został pokazany, jest wdrażanie aplikacji. Wdrożenie można zautomatyzować za pomocą Jenkins, który jest narzędziem do ciągłej integracji i ciągłego wdrażania (CI/CD). Jenkins automatycznie buduje i testuje kod po każdym commitcie, co zapewnia, że błędy są wykrywane na wczesnym etapie.

Przykład konfiguracji pipeline'u w Jenkinsie dla projektu C++:

```
pipeline {
  agent any
  stages {
    stage('Build') {
      steps {
        sh 'cmake .'
        sh 'make'
      }
    }
    stage('Test') {
      steps {
        sh './MyAppTest'
      }
    }
    stage('Deploy') {
      steps {
        echo 'Wdrażanie aplikacji...'
            }
    }
  }
```

}

W powyższym pipeline Jenkins wykonuje kroki budowania i testowania aplikacji, a następnie wdrażania jej na środowisko produkcyjne