

tesi di laurea magistrale

DevOps : sviluppo di una toolchain in ambito C++

Anno Accademico 2020/2021

relatore

Ch.mo Prof Porfirio Tramontana

correlatore

Ing. Carmelo Mirabile

candidato

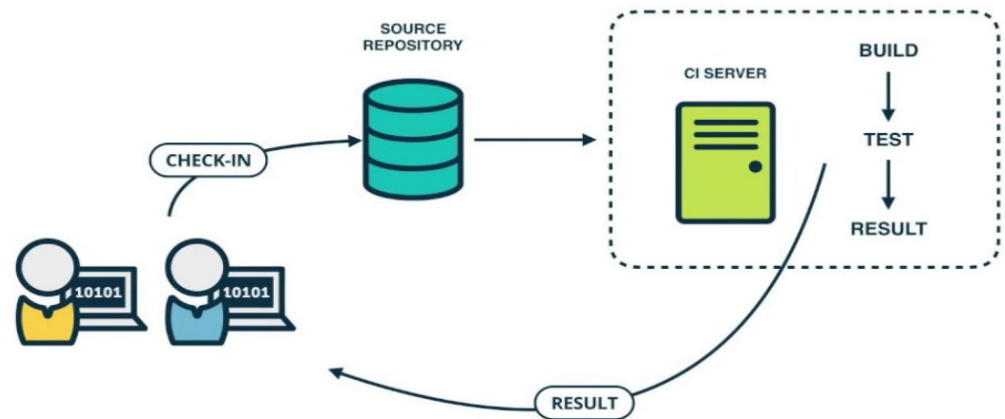
Fernando Di Costanzo

Matr. M63000925

Introduzione

- **Integrazione frequente** del lavoro
- Ogni integrazione è verificata da compilazione ed altre attività **automatiche**
- Si **riducono** i problemi legati all'integrazione

Continuous Integration (CI)



- **L'obiettivo** finale di questa tesi è l'implementazione di una **toolchain DevOps**, attraverso la quale è possibile ottenere la **Continuous Integration (CI)**.

DevOps : Development & Operation (1/2)

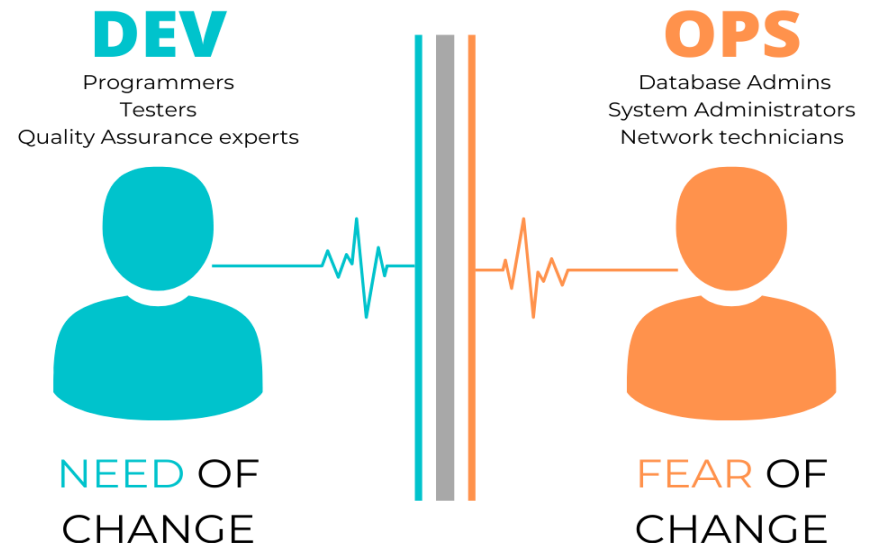
- Estende i principi **Agile** enfatizzando gli strumenti
- **Promuove** la stretta **collaborazione** tra le due divisioni
- Enfatizza l'uso **dell'automazione** nelle varie fasi

Dev ed Ops hanno lo stesso obiettivo : l'aumento della produttività all'interno dell'azienda al fine di migliorare i profitti.

Tuttavia... **Lo sviluppo tende al cambiamento** mentre **l'operation lo teme.**

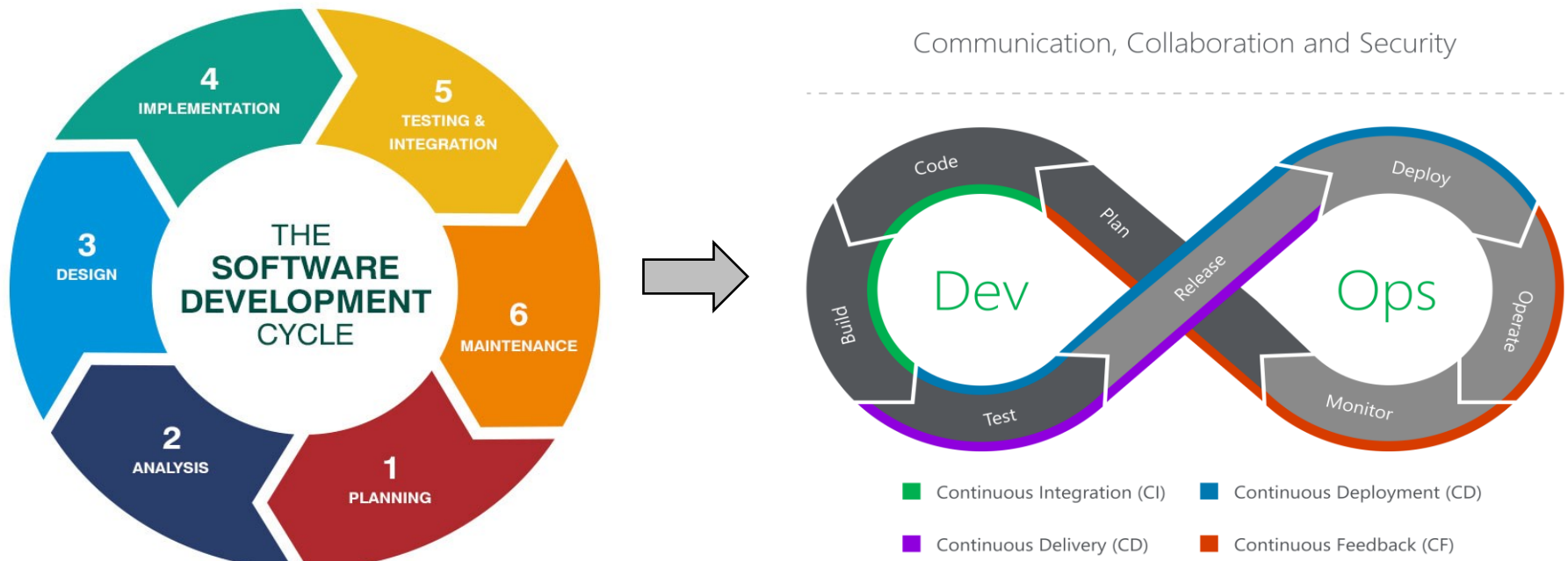
- Miglioramento interfaccia
- Correzione Bug
- Nuove features

- Stabilità
- Robustezza
- Disponibilità



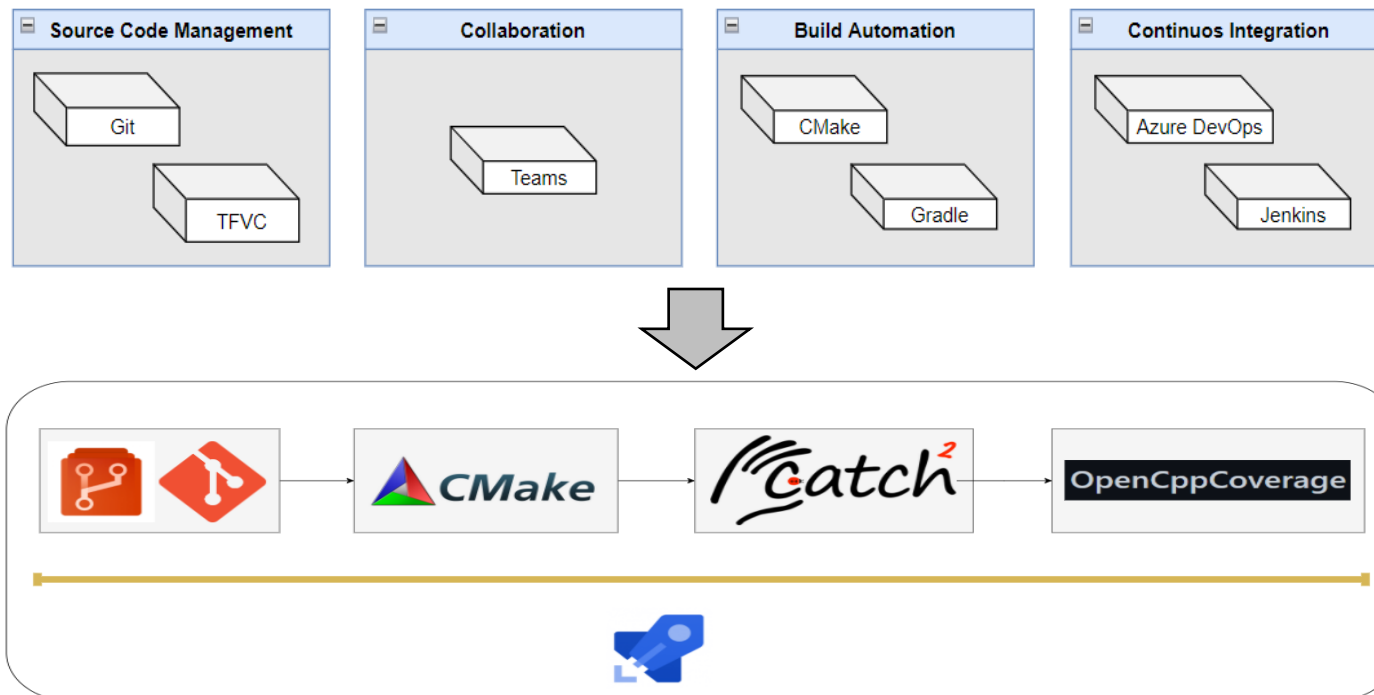
DevOps : Development & Operation (2/2)

➤ DevOps influenza l'intero ciclo di vita del software...



Definizione e Realizzazione di una toolchain (1/2)

- **Combinazione degli strumenti**
- **Compatibilità** tra gli strumenti;
- Due categorie : **In-The-Box** e **Custom**.

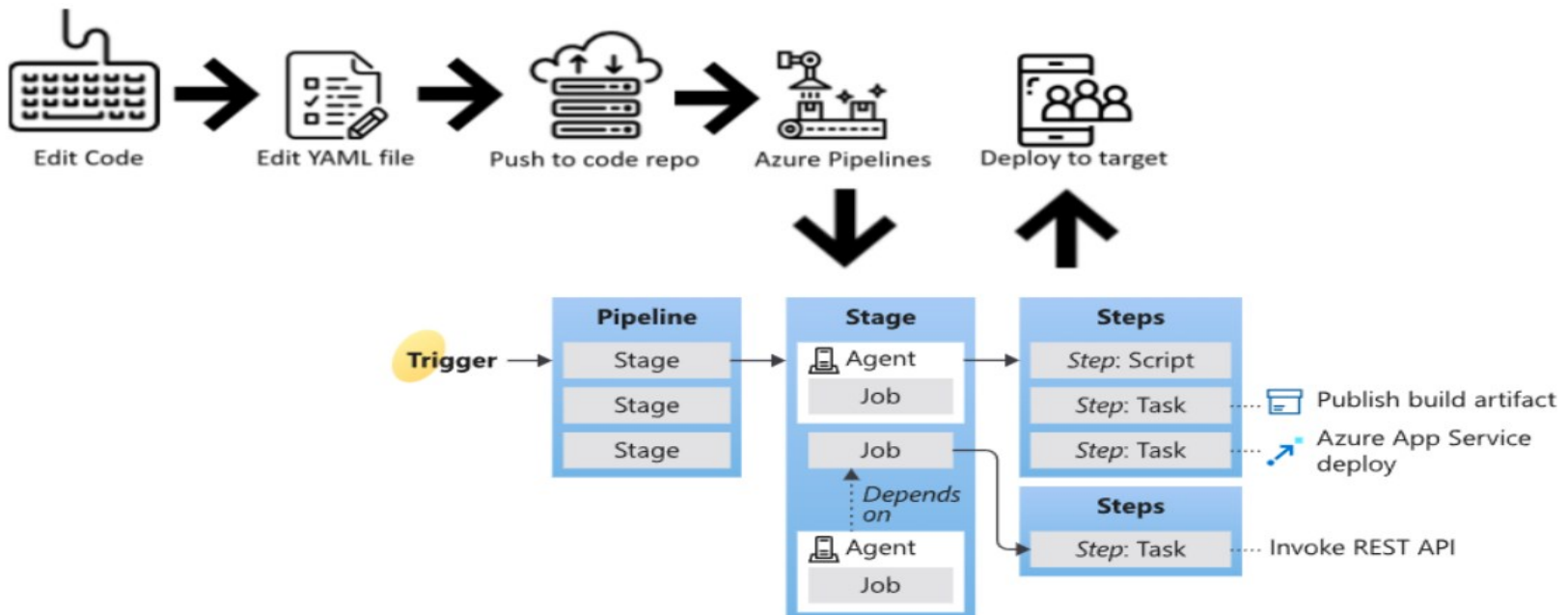


Definizione e Realizzazione di una toolchain (2/2)

➤ Azure Pipeline

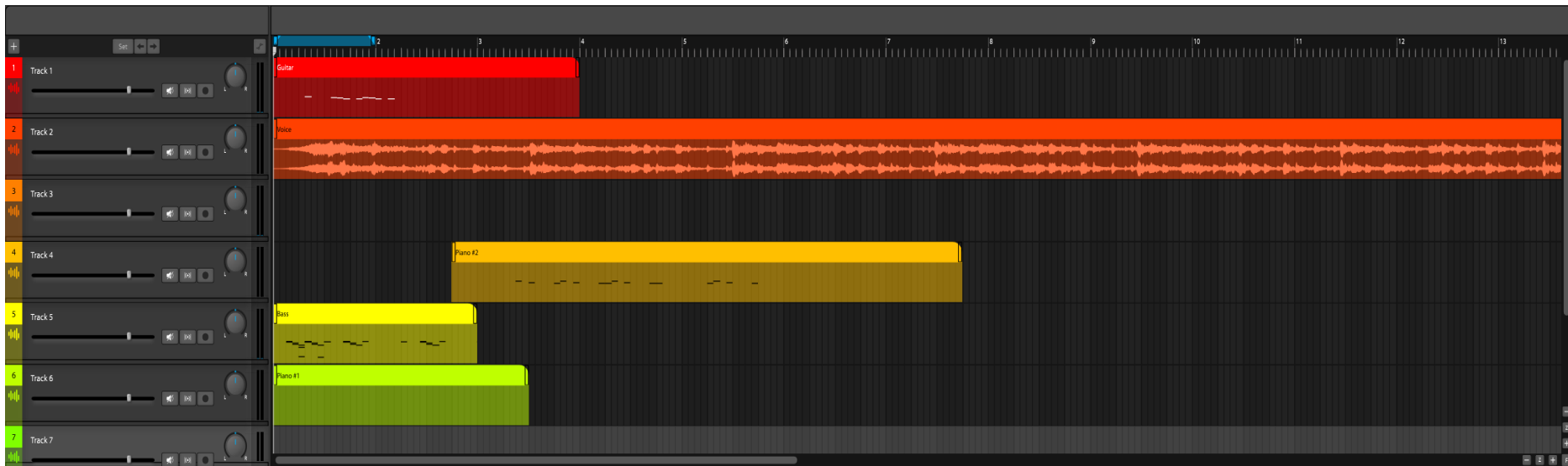
Offre servizi di compilazione/rilascio per supportare l'**integrazione** e la **delivery** continua :

- Configurazione di Azure Pipelines per l'uso del repository remoto.
- Creazione e configurazione del file ***azure-pipelines.yaml*** per definirne il comportamento
- Configurazione del trigger per l'attivazione della pipeline



Caso di Studio (1/4)

Sviluppo di una D.A.W. (Digital Audio Workstation)

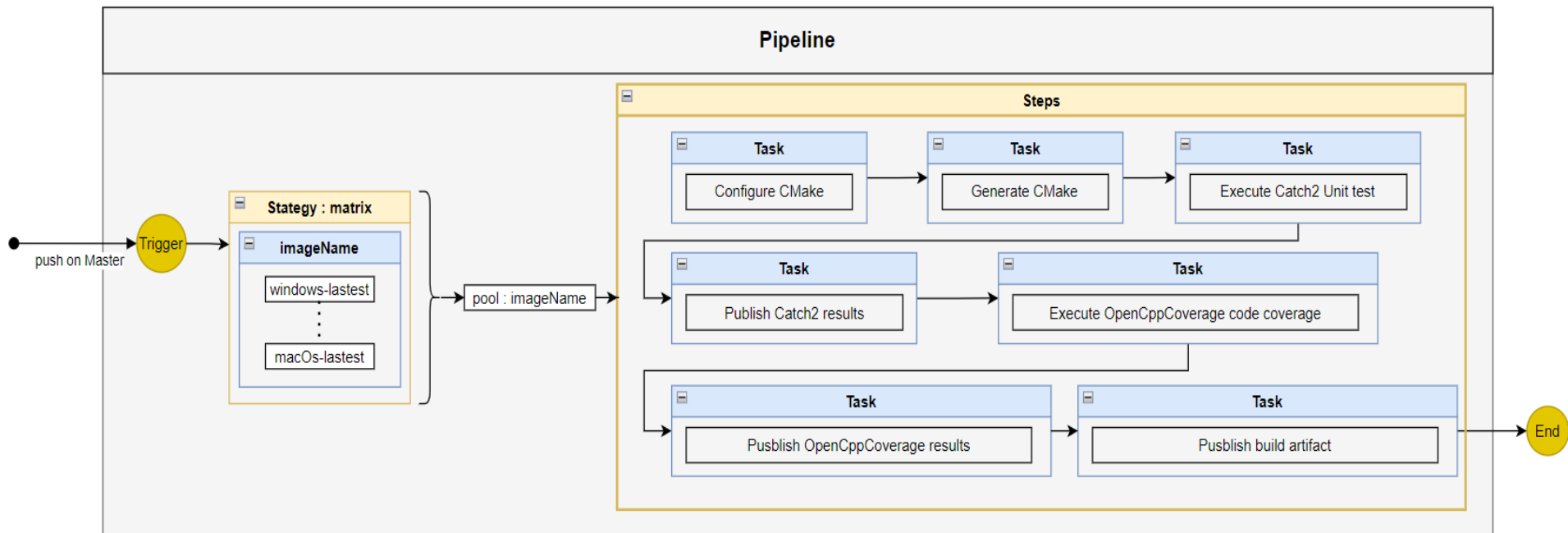


Uso del **C++** come vincolo tecnologico :

- Presenza di librerie/framework a supporto per l'audio programming più efficienti
- Più efficace per la programmazione a livello di sistema

Caso di Studio (2/4)

➤ Configurazione pipeline



Caso di Studio (3/4)

```
- task: CMake@1
  displayName: 'CMake ..'
  inputs:
    workingDirectory: 'cmake-build'
    cmakeArgs: '.. -D BUILD_UNIT_TESTS=ON'
```

```
- task: CMake@1
  displayName: 'CMake --build . --config Debug'
  inputs:
    workingDirectory: 'cmake-build'
    cmakeArgs: '--build . --config Debug'
```



Configurazione e
generazione della build
tramite **CMake**

```
- task: BatchScript@1
  displayName: 'Executing Catch2 Unit Tests...'
  inputs:
    filename: 'cmake-build\tests_catch\Catch2_UnitTestRunner_artefacts\
Debug\Catch2_UnitTestRunner.exe'
    arguments: '-r junit > $(System.DefaultWorkingDirectory)\report.xml'
    modifyEnvironment: true
    enabled: true
    continueOnError: true
```



Esecuzione degli Unit
Test con **Catch2**

```
- task: PublishTestResults@2
  inputs:
    testResultsFormat: 'JUnit'
    testResultsFiles: '$(System.DefaultWorkingDirectory)\report.xml'
    failTaskOnFailedTests: false
    displayName: 'Publishing test results...'
```



Pubblicazione risultati
test

Caso di Studio (4/4)

```
- task: BatchScript@1
  displayName: 'Getting code coverage...'
  inputs:
    filename: '3rdparty\OpenCppCoverage\OpenCppCoverage.exe'
    arguments: '--sources=modules\dawtribe_components
               --export_type=cobertura:$(System.DefaultWorkingDirectory)\codeCovOut\
cobertura.xml
               cmake-build\tests_catch\Catch2_UnitTestRunner_artefacts\Debug\
Catch2_UnitTestRunner.exe '
    modifyEnvironment: true
    enabled: true
    continueOnError: true
```



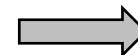
Valutazione della code
coverage con
OpenCppCoverage

```
- task: PublishCodeCoverageResults@1
  inputs:
    codeCoverageTool: 'Cobertura'
    summaryFileLocation: $(System.DefaultWorkingDirectory)\codeCovOut\
cobertura.xml
```



Pubblicazione risultati code
coverage

```
- task: PublishBuildArtifacts@1
  displayName: 'Publish Artifact: drop'
```



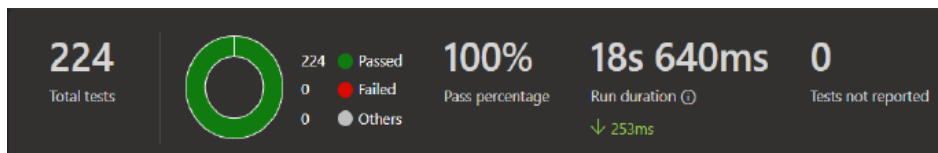
Pubblicazione degli artefatti di
build

Risultati (1/2)

➤ Risultati cumulativi

- Pipeline pass rate
- Test pass rate
- Pipeline duration

➤ Risultati di dettaglio



Coverage

Name	Covered	Uncovered	Coverable	Total	Line coverage	Branch coverage
D:\a\1\s\cmake-build\tests_catch\Catch2_UnitTestRunner_artefacts\Debug\Catch2_UnitTestRunner.exe	10043	14867	15817	46623	6%	
ArrangementHeaderComponent.cpp	0	56	56	132	0%	
ArrangementHeaderComponent.h	0	6	6	62	0%	
ArrangerView.cpp	0	2	2	35	0%	
AudioClipComponent.cpp	25	117	142	411	17.6%	
AudioClipComponent.h	3	1	4	76	75%	
AudioClipComponentWrapper.cpp	33	3	36	312	91.6%	

DevOps : sviluppo di una toolchain in ambito C++

Scenario: Creating new clip constr/Given: [an audio track] && [the edit component]

Result Details

✗ Failed giovedì on tbd

Duration: 0:00:00.723
Owner: not available
Date started: 20/1/2022, 12:37:07
Date completed: 20/1/2022, 12:37:07
Failing since: giovedì
Failing since build: 20220120.1

Debug Work items Attachments History

▼ Error message

folder.exists() == true

▼ Stack trace

```

FAILED:
  REQUIRE( folder.exists() == true )
  with expansion:
    false == true
  at D:\a\1\s\tests_catch\ClipConstrainerTest.cpp(35)

```

Risultati (2/2)

- **Collaborazione più stretta mediante sistemi di notifiche.**
- **Risparmio di tempo ed effort.**

Conclusioni

- ✓ **Automatizzazione dell'integrazione** con conseguente riduzione di **tempi** e **sforzi**.
- ✓ **Acquisizione di nuove conoscenze** legate allo sviluppo di una **toolchain** e dei **tools** che la compongono.
- ✓ Utilizzo di **Azure DevOps**: essenziale per la gestione della fase di **planning** con particolare enfasi sul lato **pipeline**.

Sviluppi Futuri

- ❑ **Sviluppo della pipeline per la CD.**
- ❑ **Parallelizzazione dell'esecuzione della pipeline CI su più agent.**

Grazie per l'attenzione.

relatore

Ch.mo Prof Porfirio Tramontana

candidato

Fernando Di Costanzo

Matr. M63000925