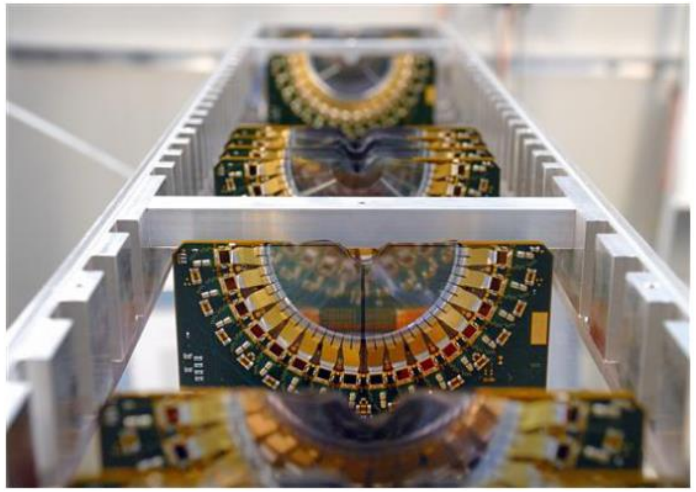
Development of core functions in test-beam data analysis software

Author: Paweł Korytowski Supervisor: dr inż. Bartłomiej Rachwał

Introduction

Silicon sensors work in heavy environment, they have to survive harsh radiation. They also have to provide a satisfactory performance throughout the experiment.

Reliability and other parameters are being tested during R&D activity at special test beam environment. We can generate big amount of data. With data analysis, we can investigate which detector will be the best in our experiment.

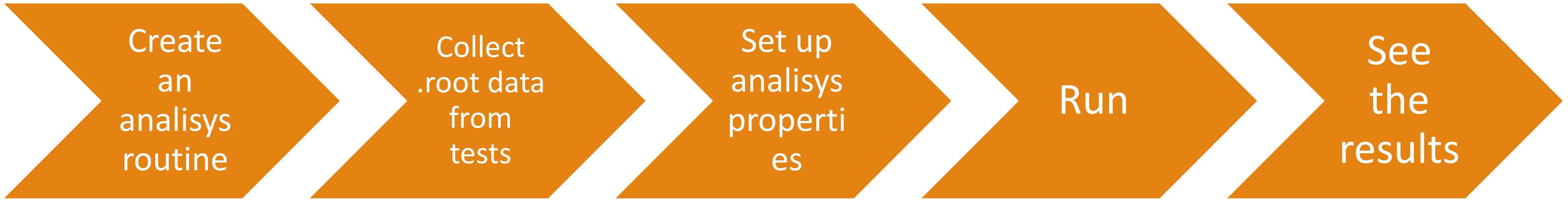

Detectors from LHCb VELO experiment

TbGaudi – what is it?

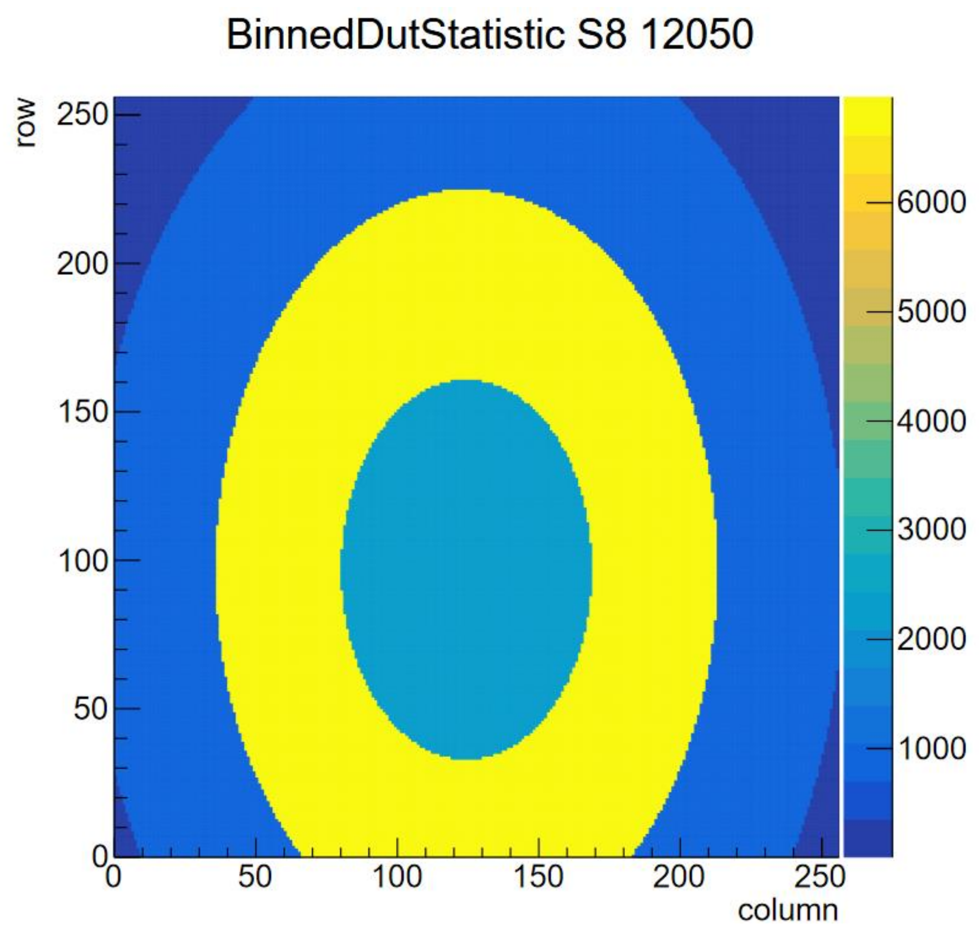
Test-beam like data represent predefined pattern/structure, so doing the very first data analysis like e.g. generating plots or histograms manually is simply waste of time. For it's automatization there is the TbGaudi package.

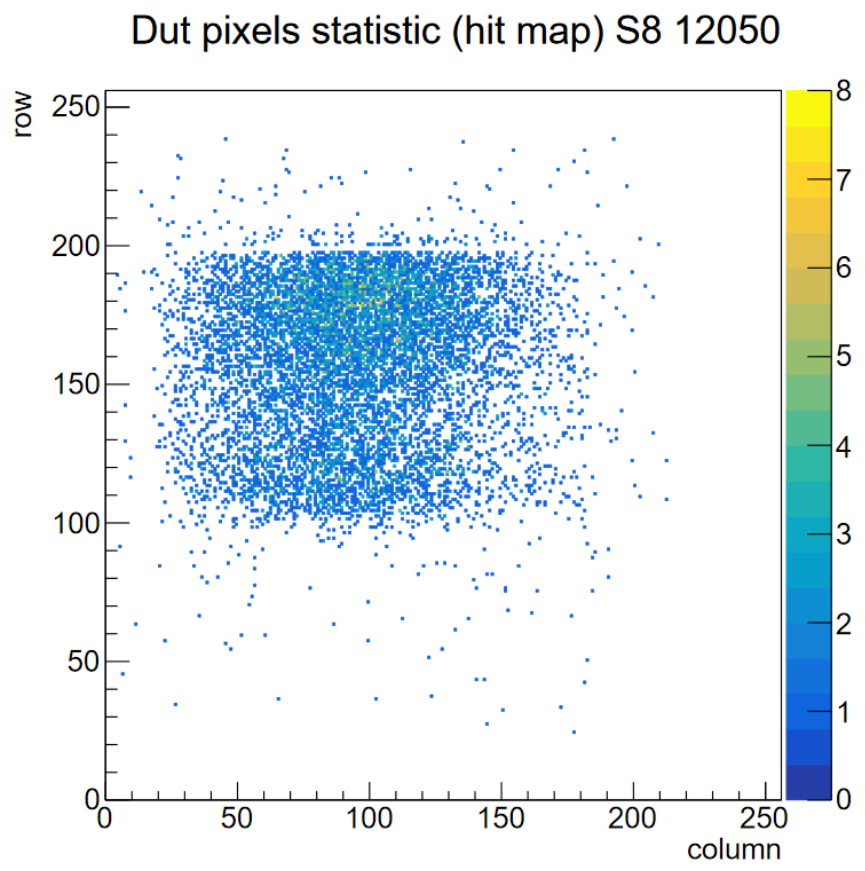
TbGaudi package is a modular software for test-beam data analysis. It automates an analysis process for different detectors or different testing conditions. Once defined analysis routine can be used multiple times with many datasets. User can change specific parameters for every run.

How it works

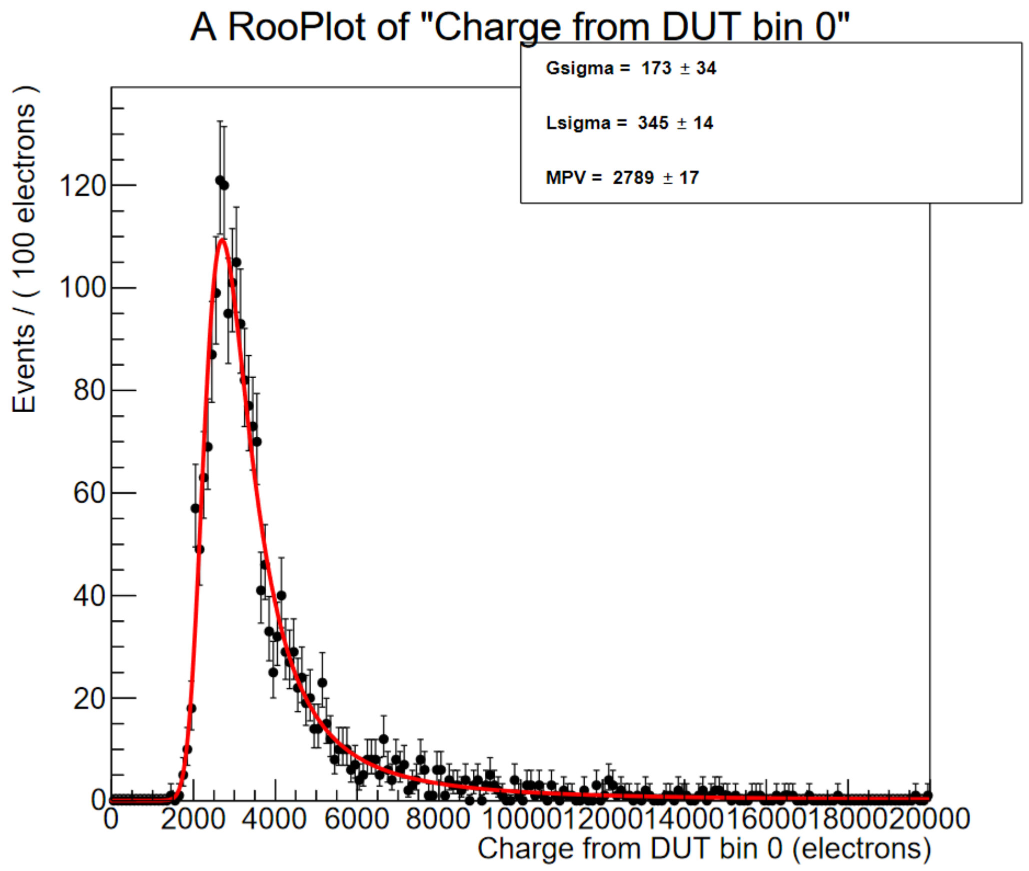

Create an analisys routine
Collect .root data from tests
Set up analisys properties
Run
See the results

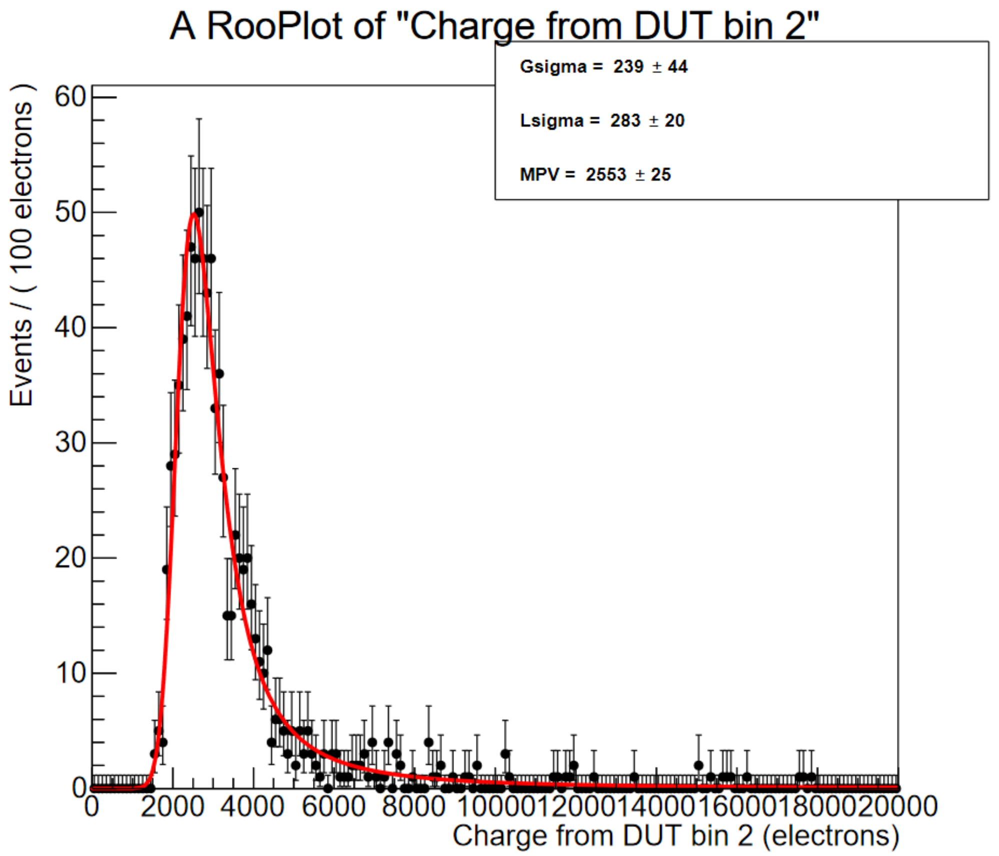
Sample results





Sample results





First task – introducing logger to the project

So far, every message displayed by the console was created like below:

```
std::cout<<„[INFO]: Some message”;
```

Why it is not the best idea?

- we have to create every message manually

- we can not redirect messages to a file

- we can create a mess (by e.g. typos)

- we have not any control (e.g. changing displayed level)

TbLog – the new logger class for TbGaudi

General principles:

- As simple as possible even if it means reduced functionality – framework is used by scientists, they don’t want to waste time learning logger stuff
- Must look like integral part of the project
- Users can define their own logging streams on demand – they want to keep some kind of information in one file
- Logging messages can be save to the file

TbLog – the new logger class for TbGaudi

I decided not to create a logger from scratch. It is like breaking through open door. I can not simply add an external library to the project because I want to keep the project consistent.

I decided to use simple and fast logging library – Spdlog[1]. It has everything I need and it is easy to add it to CMake. The project is alive so it receives updates, bug fixes, etc. It has a large community too.

I decided to create a class with essential functionality of Spdlog.

Why not Boost::Log?

Boost::Log is great but it needs a lot of dependencies and it has to be precompiled before use.

[1] https://github.com/gabime/spdlog - Spdlog library repository

TbLog – user streams

It was really important to allow users to create their own logging streams. They can be created in any place in a program. The only thing we need to do is simply write:

TbLog::AddLogger(„my_new_logger”);

Now, we can send a message to the our stream e.g. :

TbLog::Info(„my_new_logger”, „This is awesome!”);

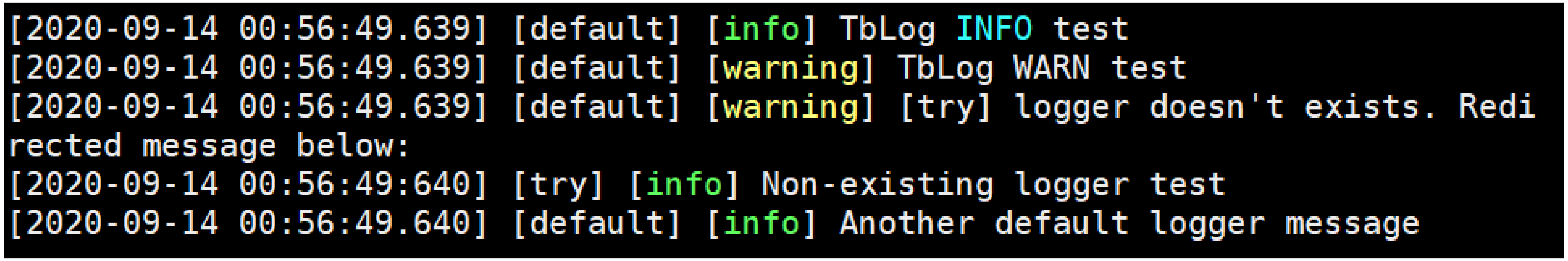
What are the benefits?

New stream creates a new file where it's message are stored. Message's preamble changes from [default] to [my_new_logger].

When a logger does not exists, messages are redirected to the default stream with a warning.

TbLog – example of usage and output

```
TbLog::Info(„TbLog INFO test”);                        //-> message type „Info”, default stream
TbLog::Warn(„TbLog WARN test);                         //-> message type „Warn”, default stream
TbLog::Info(„try”, „Non-existing logger test”);        //-> message type „Info”, user-def. stream
TbLog::Info(„Another default logger message”);  //-> message type „Info”, default stream
```



```
[2020-09-14 00:56:49.639] [default] [info] TbLog INFO test
[2020-09-14 00:56:49.639] [default] [warning] TbLog WARN test
[2020-09-14 00:56:49.639] [default] [warning] [try] logger doesn't exists. Redi
rected message below:
[2020-09-14 00:56:49:640] [try] [info] Non-existing logger test
[2020-09-14 00:56:49.640] [default] [info] Another default logger message
```

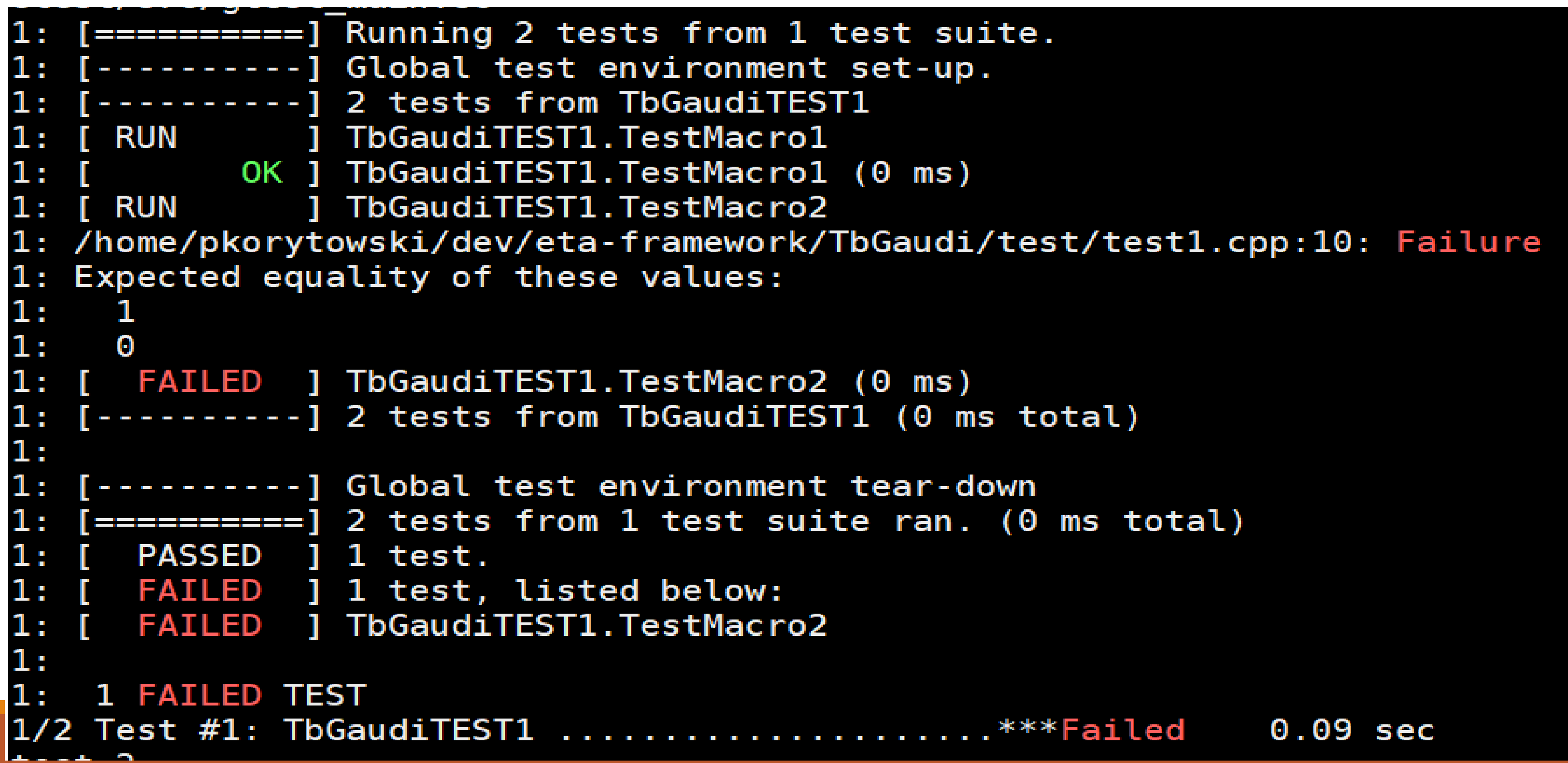
TbLog – technical details

- singleton – only one instance in a program
- All user streams are keeping in memory as shared_ptr – no need to explicitly delete it by the user
- Available log message types/levels: Trace, Debug, Info, Warn, Error, Critical
- User can change sensitivity of every stream
- Functionality can be expanded quickly, because many methods exists in Spdlog but are hidden by the interface.

Second task – introduce unit tests to the project

So far, there were no tests in the project. With that amount of data it is hard to find bugs or unwanted behavior of the app. The decision was made – tests are really needed. Now, TbGaudi has an excellent environment – Google Tests.

Here is a short example output:



```
1: [==========] Running 2 tests from 1 test suite.
1: [----------] Global test environment set-up.
1: [----------] 2 tests from TbGaudiTEST1
1: [ RUN      ] TbGaudiTEST1.TestMacro1
1: [       OK ] TbGaudiTEST1.TestMacro1 (0 ms)
1: [ RUN      ] TbGaudiTEST1.TestMacro2
1: /home/pkorytowski/dev/eta-framework/TbGaudi/test/test1.cpp:10: Failure
1: Expected equality of these values:
1:   1
1:   0
1: [  FAILED  ] TbGaudiTEST1.TestMacro2 (0 ms)
1: [----------] 2 tests from TbGaudiTEST1 (0 ms total)
1:
1: [----------] Global test environment tear-down
1: [==========] 2 tests from 1 test suite ran. (0 ms total)
1: [  PASSED  ] 1 test.
1: [  FAILED  ] 1 test, listed below:
1: [  FAILED  ] TbGaudiTEST1.TestMacro2
1:
1:  1 FAILED TEST
1/2 Test #1: TbGaudiTEST1 .....................***Failed    0.09 sec
```

Unit tests task - challenges

- Understanding the idea of unit tests
- Gaining knowledge about different testing frameworks (Google Test is not only the one)
- Integration of testing environment with an existing project using CMake
- An opportunity to deep dive in the TbGaudi framework to start writing valuable tests

Thank you for your attention