Krakow Applied Physics and Computer Science Summer School '20

ONLINE

July 13 - 24 2020
September 7 - 18 - seminar sessions

Analysis Techniques and Team Collaboration Tools

15 July 2020

Bartek Rachwał
brachwal [at] agh.edu.pl

AGH

Wydział Fizyki i Informatyki Stosowanej

LHCb

ATLAS Experiment

Intelli-NET

The scope of the lecture

Part 1: Team Collaboration Tool(s)

Effective team workflow(s)

Part 2: Analysis Techniques

ROOT introduction, basics, tips and tricks

The scope of the lecture

Part 1: Team Collaboration Tool(s)

Effective team workflow(s)

➔ define and visualise the process

➔ Atlassian ecosystem

➔ git, boards, docs

➔ short hands-on, how it can works...

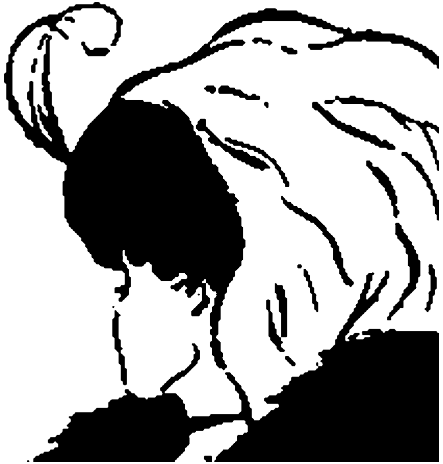
➔ alternatives?

Define and visualise the process

- ➔ **Creative/analytical problem solving** and innovation in a team environment is incredibly ambiguous/complex by nature.

Define and visualise the process

- ➔ **Creative/analytical problem solving** and innovation in a team environment is incredibly ambiguous/complex by nature.



Do you see an old woman or a young lady?

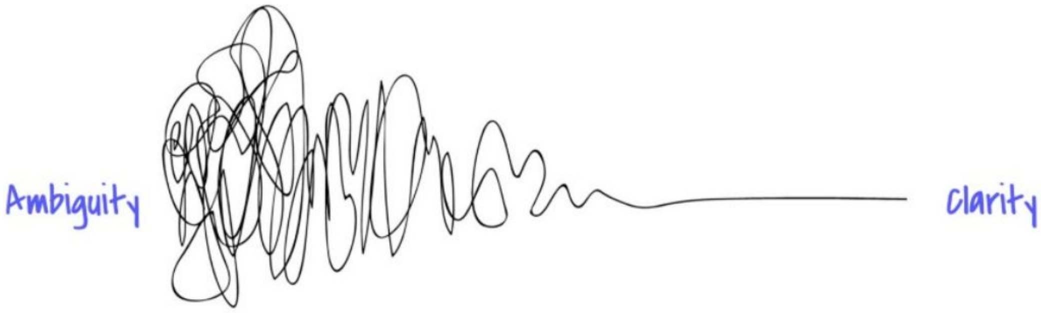
www.menti.com
917435

Define and visualise the process

- **Creative/analytical problem solving** and innovation in a team environment is incredibly ambiguous/complex by nature.

- We are trying to **make sense of complexity** — to simplify and connect the dots.

- The idea of having **a linear, repeatable process** provides us with a bit of comfort and clarity as we walk into a cloud of ambiguity/complexity **full of information** and **different ideas**.



Define and visualise the process

Align on where to start
Establish goals & outcomes
Define project guardrails
Discuss what we know (and don't)

Scoping discussions enable us to:

➔ Align on where to start
A mutually agreed upon **place to start for the project** or task at hand.

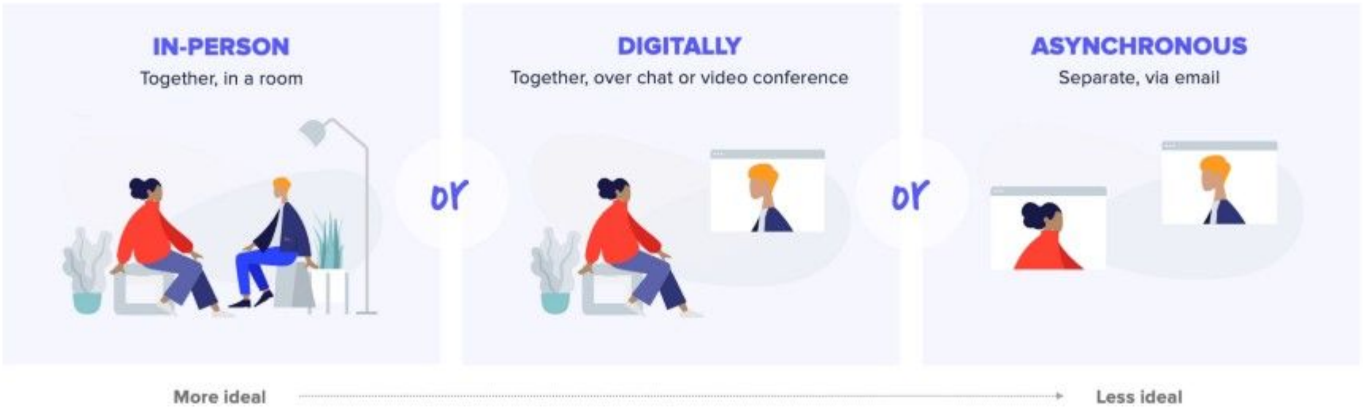
➔ Establish goals & outcomes
A mutually agreed upon definition of **what success looks like for a project** or task.

➔ Define project guardrails
A mutually agreed upon definition of what is **in scope and what is out-of-scope** for a project or task.

➔ Discuss what we know/don't know
An understanding of **what we know, and what we don't know,** about the problem we are trying to solve.

Define and visualise the process

- Align on where to start
- Establish goals & outcomes
- Define project guardrails
- Discuss what we know (and don't)



IN-PERSON
Together, in a room

or

DIGITALLY
Together, over chat or video conference

or

ASYNCHRONOUS
Separate, via email

More ideal → Less ideal

Define and visualise the process

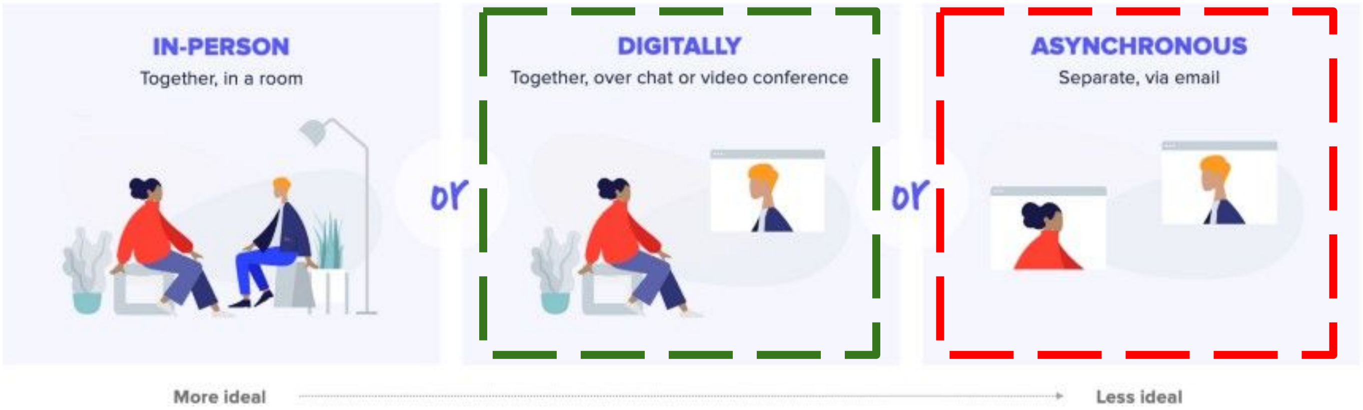
Align on where to start

Establish goals & outcomes

Define project guardrails

Discuss what we know (and don't)

What we can do nowadays...



IN-PERSON
Together, in a room

or

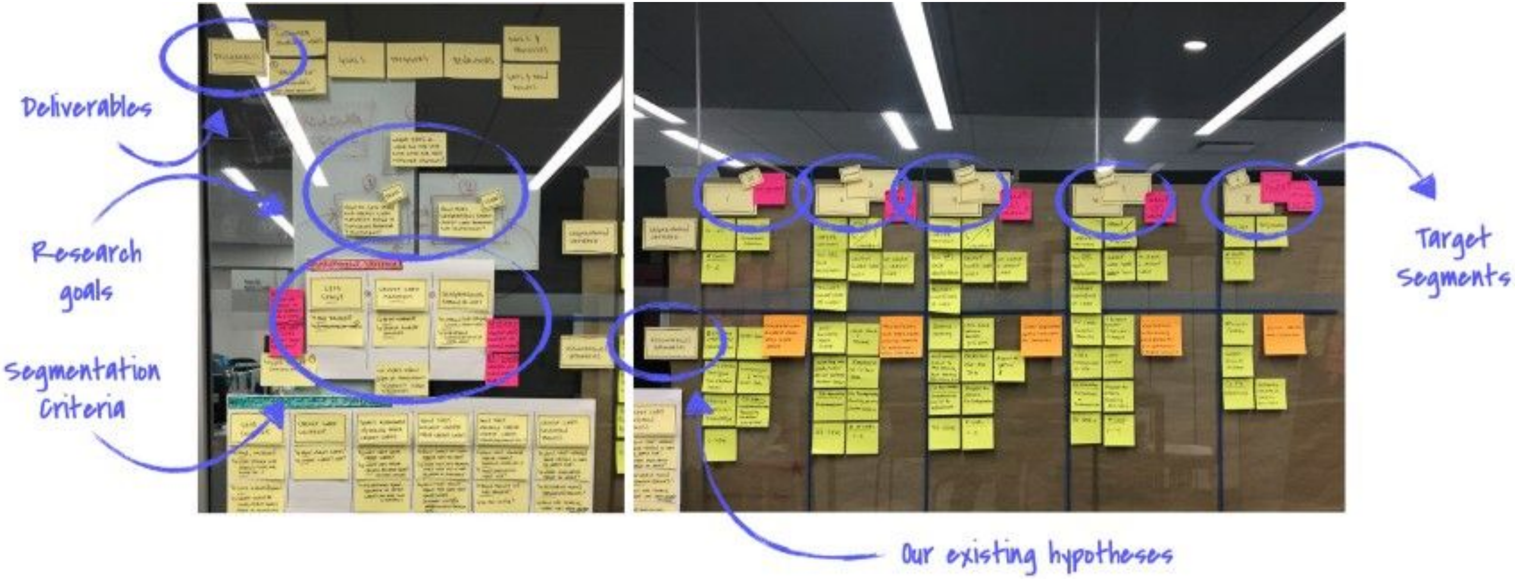
DIGITALLY
Together, over chat or video conference

or

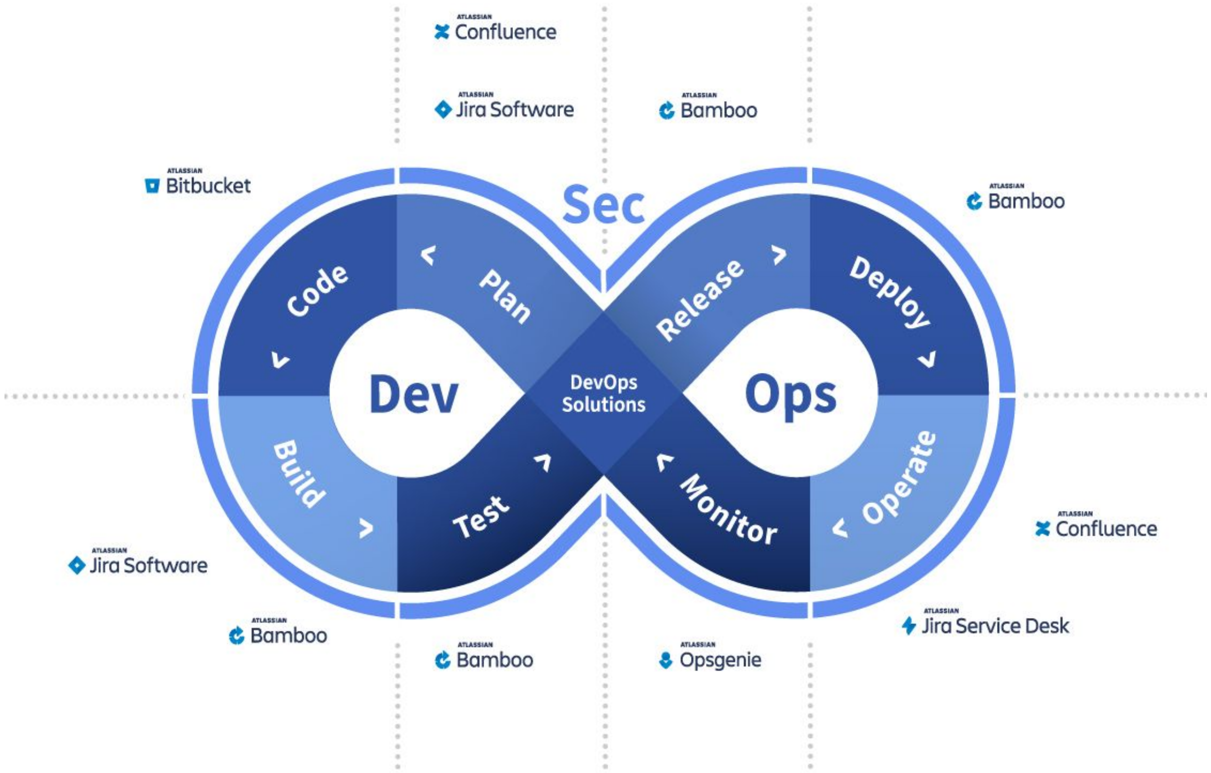
ASYNCHRONOUS
Separate, via email

More ideal → Less ideal

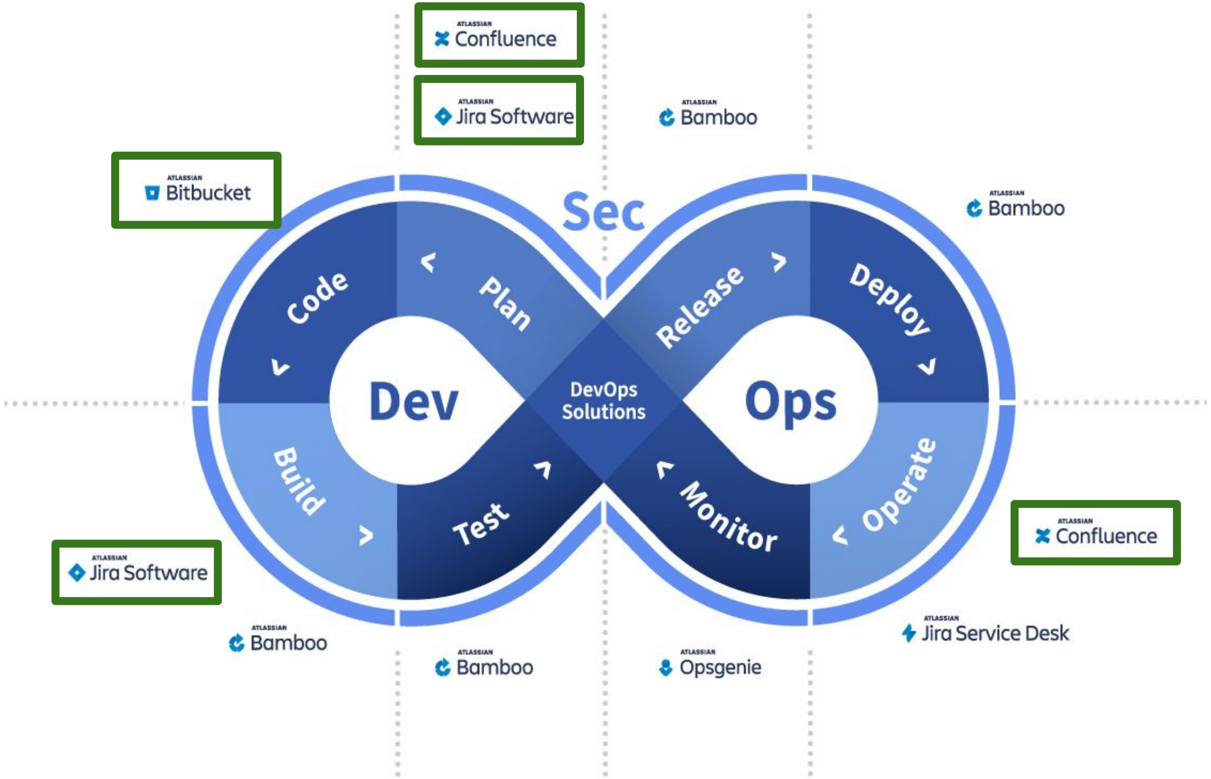
Define and **visualise the process**



Atlassian ecosystem



Atlassian ecosystem

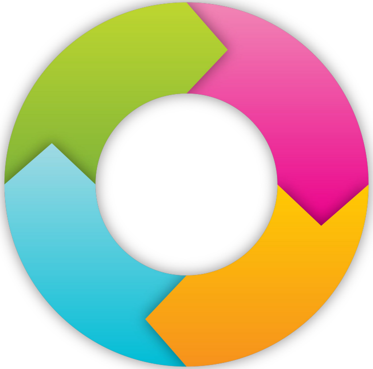


boards, git, docs

Jira

The name is actually inherited from the Japanese word "Gojira" which means "Godzilla"

➔ issue tracking
➔ bug tracking
➔ project management



Bitbucket

Place to manage git repositories, collaborate on your source code and guide you through the development flow

➔ Access control to restrict access to your source code.
➔ Workflow control to enforce a project or team workflow.
➔ Pull requests with in-line commenting for collab on code review.

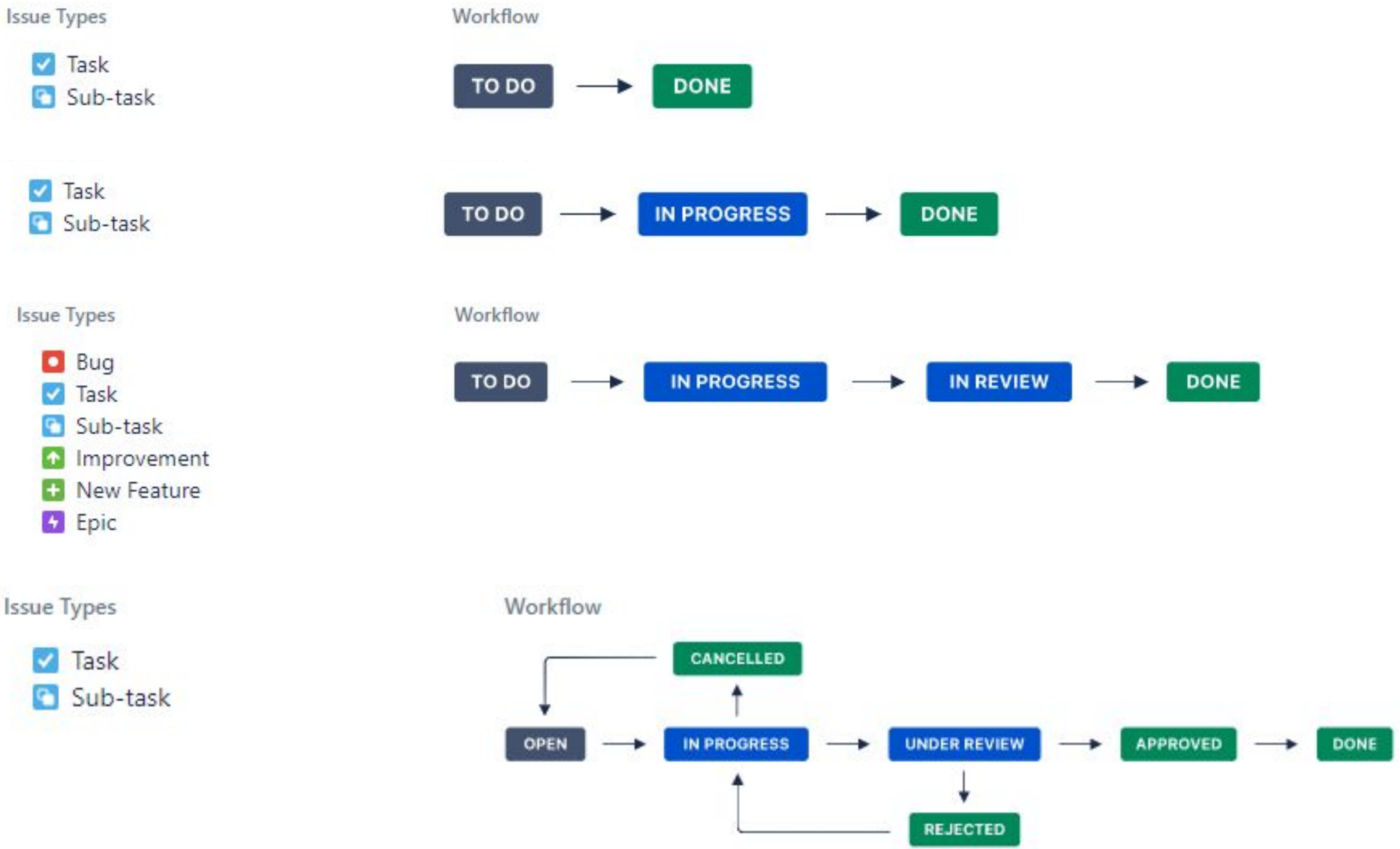
Confluence

Create, collaborate, and organize all your work in one place.

➔ great to collect keep documentation

boards, git, docs

Jira



Issue Types

- Task
- Sub-task

Workflow

TO DO → DONE

- Task
- Sub-task

TO DO → IN PROGRESS → DONE

Issue Types

- Bug
- Task
- Sub-task
- Improvement
- New Feature
- Epic

Workflow

TO DO → IN PROGRESS → IN REVIEW → DONE

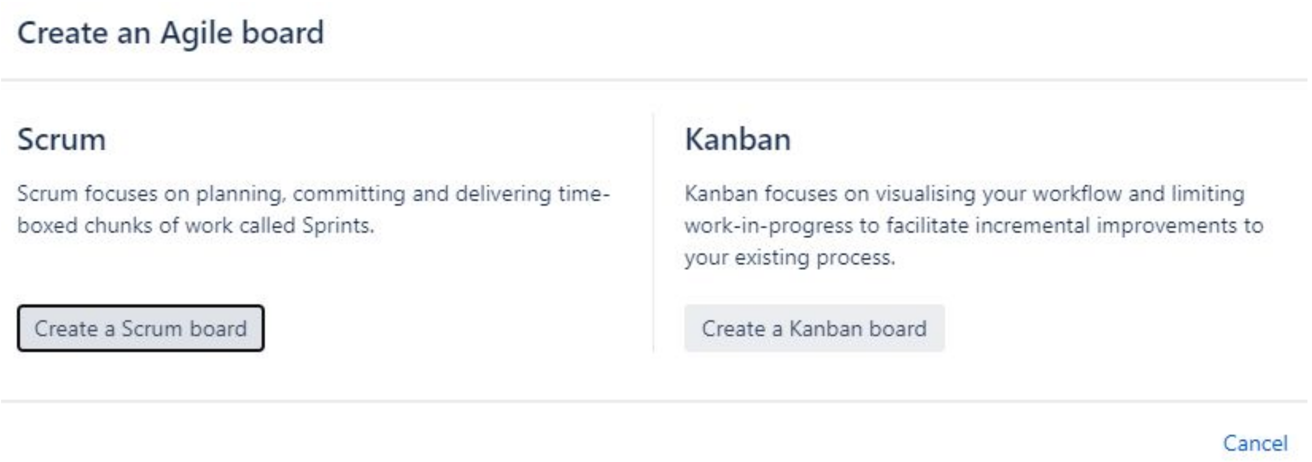
Issue Types

- Task
- Sub-task

Workflow

boards, git, docs

Jira



Create an Agile board

Scrum

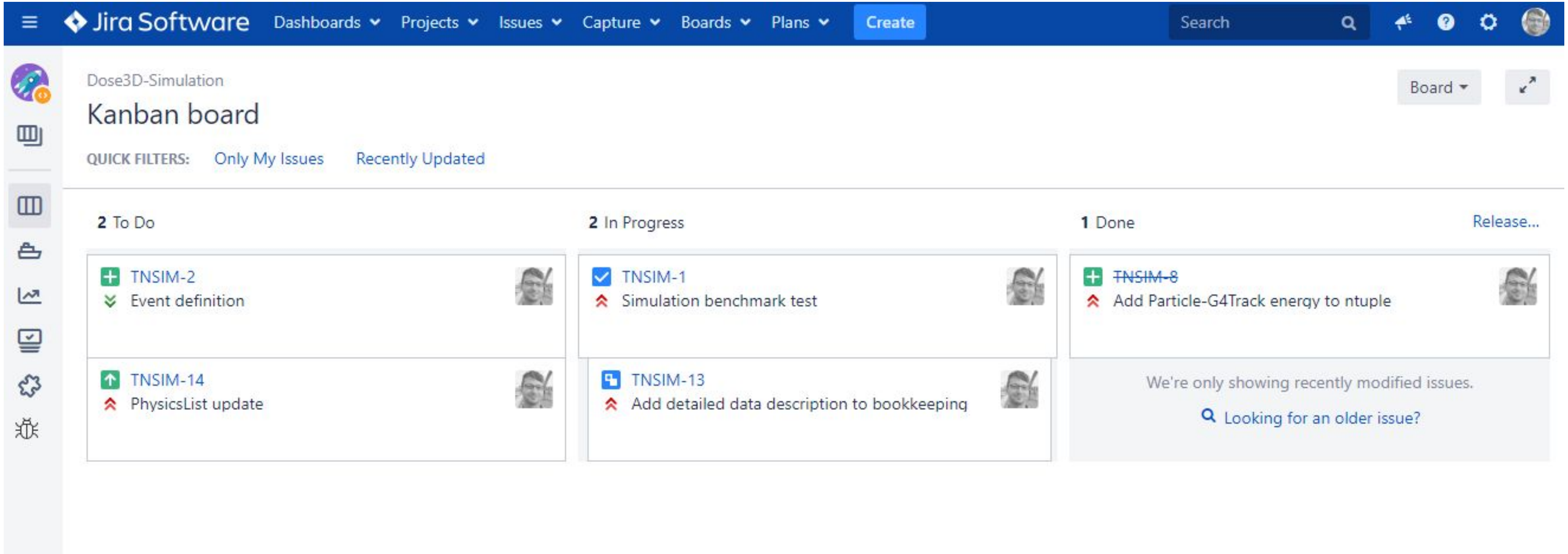
Scrum focuses on planning, committing and delivering time-boxed chunks of work called Sprints.

Kanban

Kanban focuses on visualising your workflow and limiting work-in-progress to facilitate incremental improvements to your existing process.

boards, git, docs

Jira



Dose3D-Simulation

Kanban board

QUICK FILTERS: Only My Issues Recently Updated

2 To Do

- TNSIM-2 Event definition
- TNSIM-14 PhysicsList update

2 In Progress

- TNSIM-1 Simulation benchmark test
- TNSIM-13 Add detailed data description to bookkeeping

1 Done

Release...

- ~~TNSIM-8~~ Add Particle-G4Track energy to ntuple

We're only showing recently modified issues.

Looking for an older issue?

short **hands-on**, how it can works...

- open jira task
- change status and doc page
- repo branch and pull request with code review
- close the task

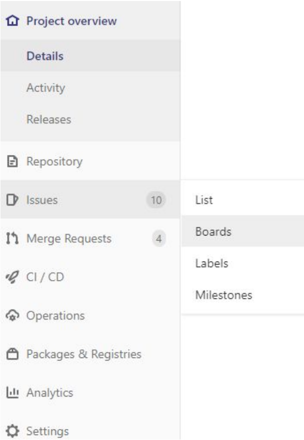
alternatives?

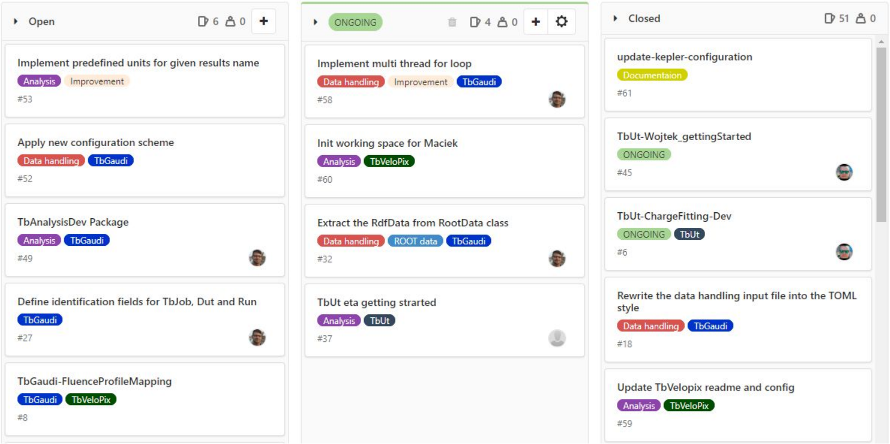
- ➔ Self-hosted or on the cloud
- ➔ Project or code dev management

GitLab

GitHub

Trello





Open 6 0

- Implement predefined units for given results name — Analysis, Improvement — #53
- Apply new configuration scheme — Data handling, TbGaudi — #52
- TbAnalysisDev Package — Analysis, TbGaudi — #49
- Define identification fields for TbJob, Dut and Run — TbGaudi — #27
- TbGaudi-FluenceProfileMapping — TbGaudi, TbVeloPix — #8

ONGOING 4 0

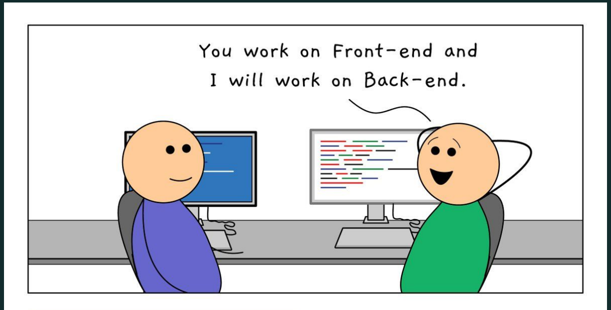
- Implement multi thread for loop — Data handling, Improvement, TbGaudi — #58
- Init working space for Maciek — Analysis, TbVeloPix — #60
- Extract the RdfData from RootData class — Data handling, ROOT data, TbGaudi — #32
- TbUt eta getting strarted — Analysis, TbUt — #37

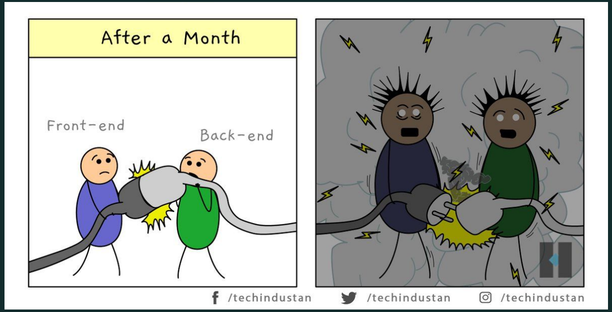
Closed 51 0

- update-kepler-configuration — Documentaion — #61
- TbUt-Wojtek_gettingStarted — ONGOING — #45
- TbUt-ChargeFitting-Dev — ONGOING, TbUt — #6
- Rewrite the data handling input file into the TOML style — Data handling, TbGaudi — #18
- Update TbVelopix readme and config — Analysis, TbVeloPix — #59

The instructions show you the way, but it won't go for you...

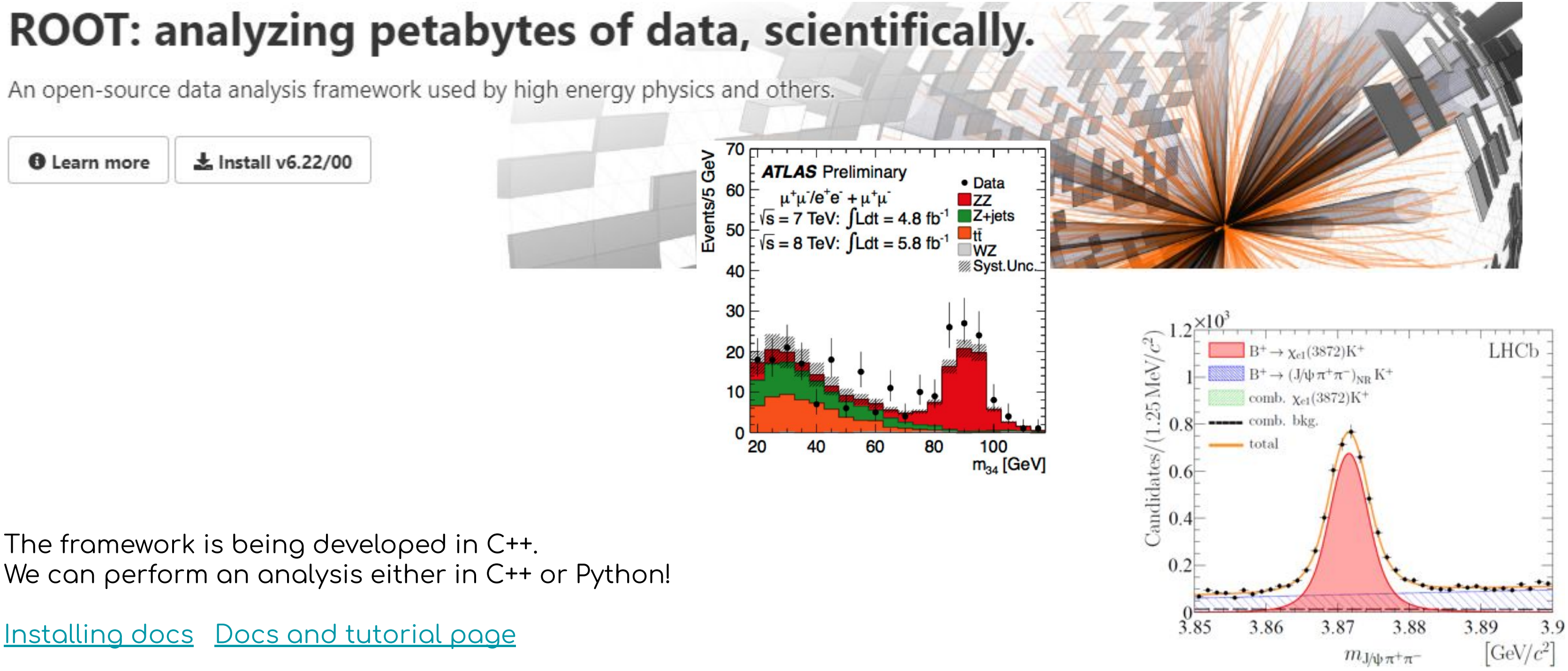
Find best env for yours (and supervisor's / teammates) needs!





The scope of the lecture

Part 2: Analysis Techniques



ROOT: analyzing petabytes of data, scientifically.

An open-source data analysis framework used by high energy physics and others.

The framework is being developed in C++.
We can perform an analysis either in C++ or Python!

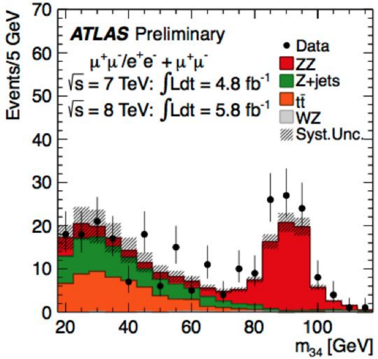
Installing docs Docs and tutorial page

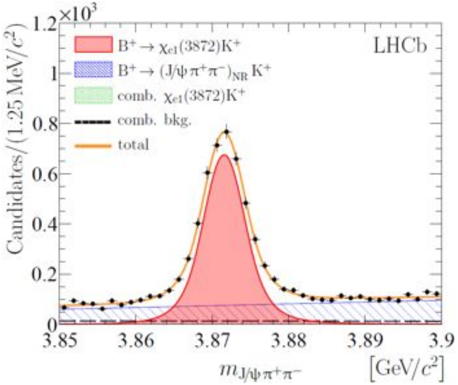
The scope of the lecture

Part 2: Analysis Techniques

ROOT introduction: basics, tips and tricks

- ➔ working interactively
- ➔ scripting with C++
- ➔ scripting with Python
- ➔ TMVA
- ➔ RooFit





Get (Py)ROOT to your desktop

Once we work with Conda… see here.

Install and run within the current conda environment

```
conda install root -c conda-forge
```

Alternatively, you can always build ROOT with cmake

Install instructions, docs

Setup ROOT in your environment:

```
$ source <installdir>/bin/thisroot.sh # or thisroot.{csh,sh} depending on the Shell
```

ROOT basics: Data structure: the TTree / NTuple

See detailed tutorial description, Trees

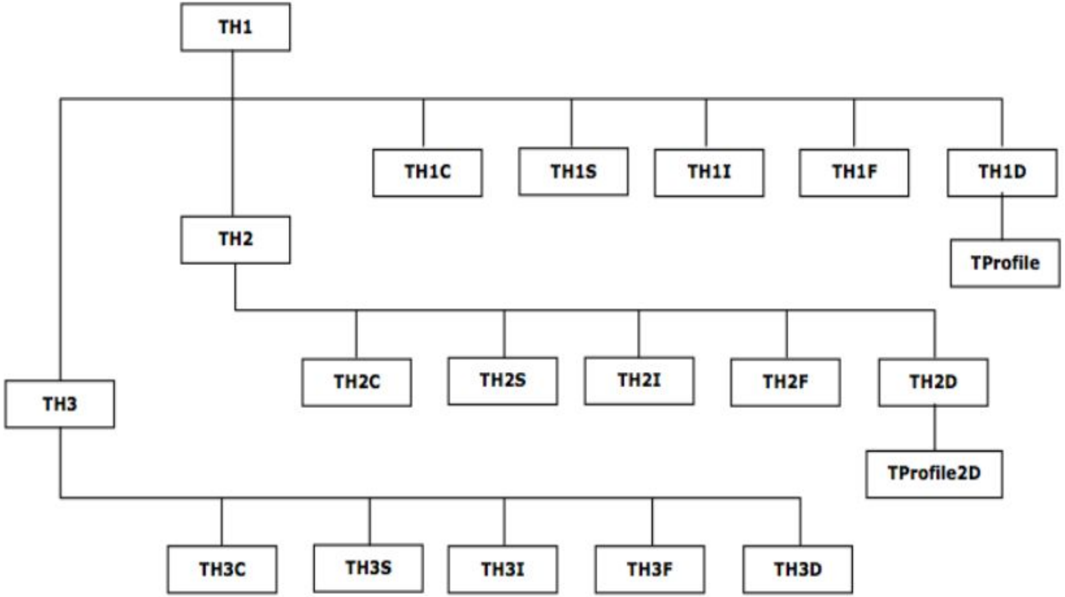
- ➔ Data is generally stored as a TTree / TNtuple
  similar to a table with rows and columns
- ➔ Each row represents an event
- ➔ Each column represents a quantity
- ➔ Trees can be created from ASCII files
- ➔ TTree class is optimized to reduce disk space and enhance access speed
- ➔ TTree can hold all kind of data

```
T->Print();    // Print the Root Content
T->Scan();     // Scan the Root rows and columns
T->Draw(“x”);  // Draw a branch of tree
T->Draw(“sqrt(x*x+y*y)”);  // Plot calculated quantity
T->Draw(“x>>h1”);          // Dump a root branch to a histogram
```

ROOT basics: Histograms

See detailed tutorial description, Histograms

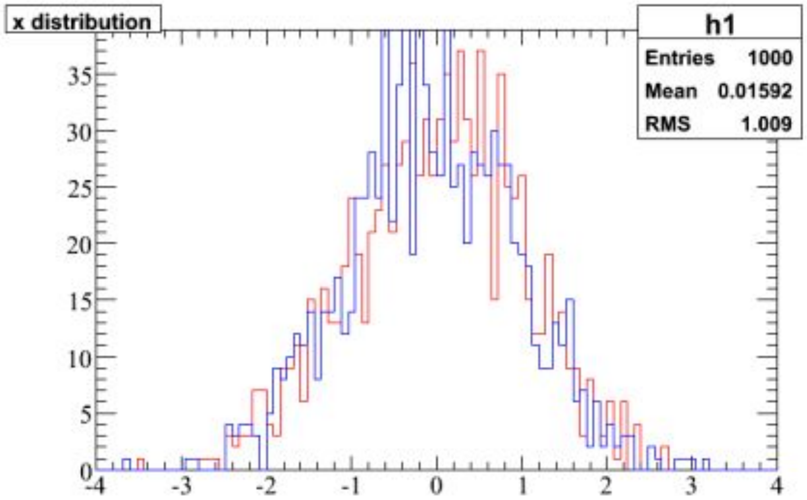
- ➔ The way we handle the data
- ➔ The inter-relation of two measured quantities X and Y can be visualized with a two-dimensional histogram or scatter-plot
- ➔ TProfile vs TProjection
  See root-forum question, link

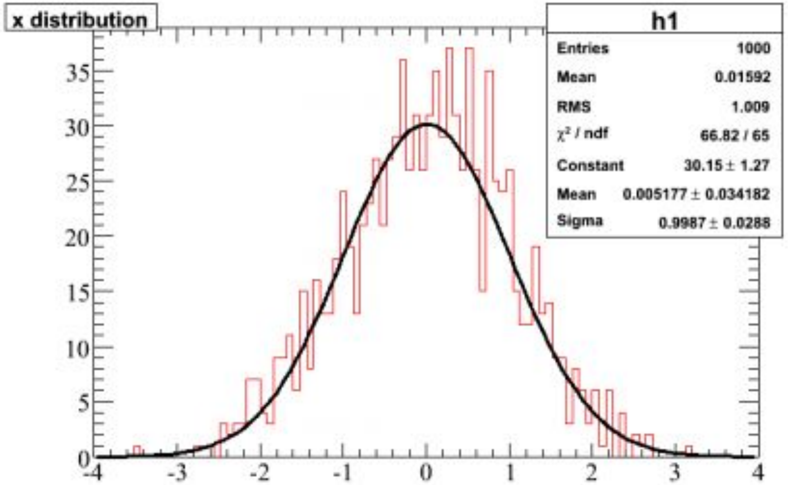


ROOT basics: Histograms

See detailed tutorial description, Histograms

- ➔ Overlapping histograms
- ➔ Simple fitting

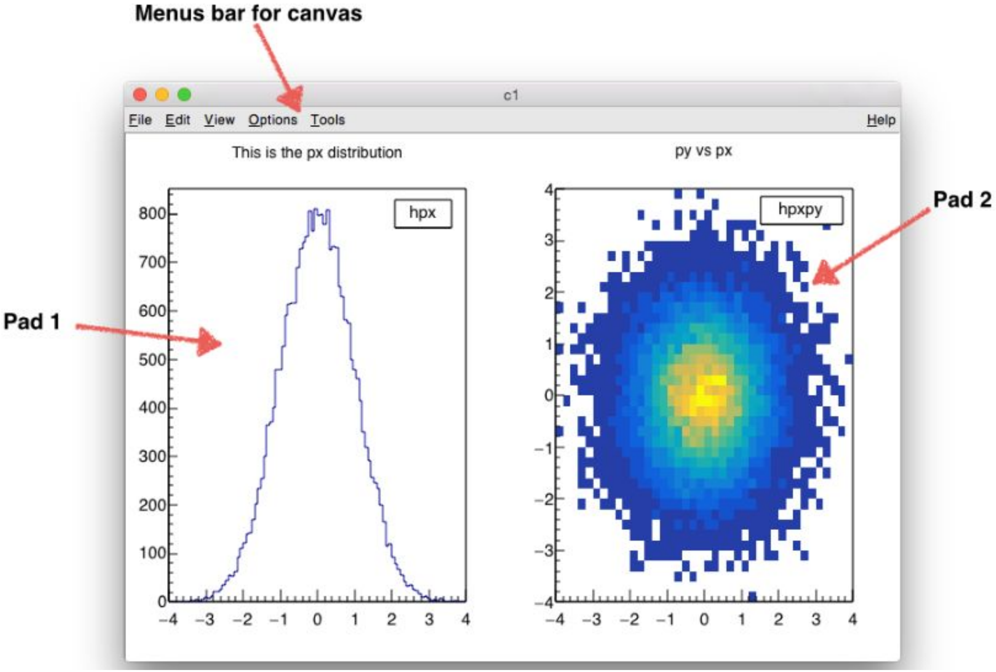




ROOT basics: Canvas

See detailed docs description, TCanvas

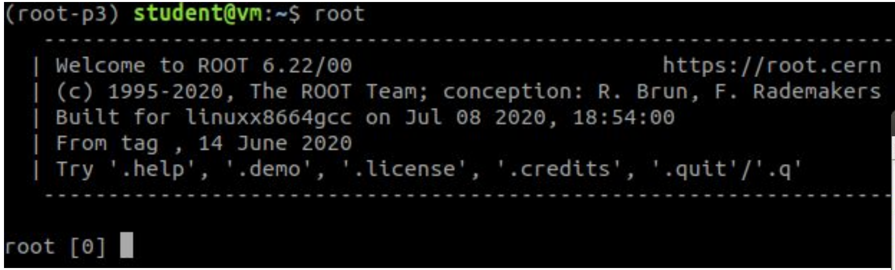
- ➔ An area mapped to a window



ROOT interactive mode

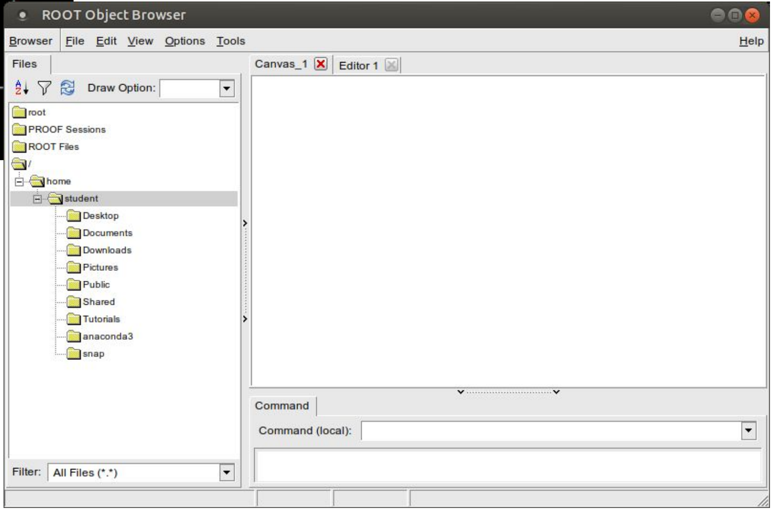
- ➔ ROOT has a C++ interpreter called cling built in.
  It is used for the prompt, both C++ and Python.

- ➔ Terminal with interactive session vs. TBrowser



```
(root-p3) student@vm:~$ root
   ------------------------------------------------------------------
  | Welcome to ROOT 6.22/00                        https://root.cern |
  | (c) 1995-2020, The ROOT Team; conception: R. Brun, F. Rademakers |
  | Built for linuxx8664gcc on Jul 08 2020, 18:54:00                 |
  | From tag , 14 June 2020                                          |
  | Try '.help', '.demo', '.license', '.credits', '.quit'/'.q'       |
   ------------------------------------------------------------------

root [0] 
```



- ➔ Let's see how it works...

ROOT C++ scripting: macros

- ➔ See official descriptions

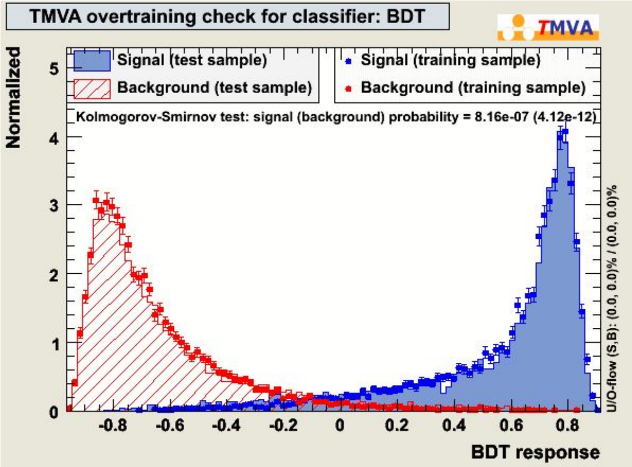
  9.2 Feeding Sources Files To ROOT: C++ Scripts, here

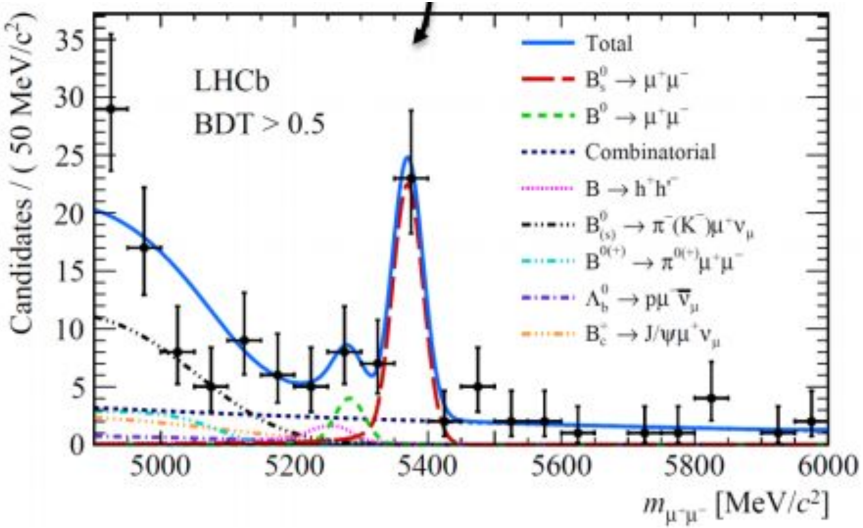
  3 ROOT Macros, here

ROOT TMVA

See tutorials, TMVA

➔ Multi-variate analysis tools including an artificial neural network and many other advanced tools for classification problems.

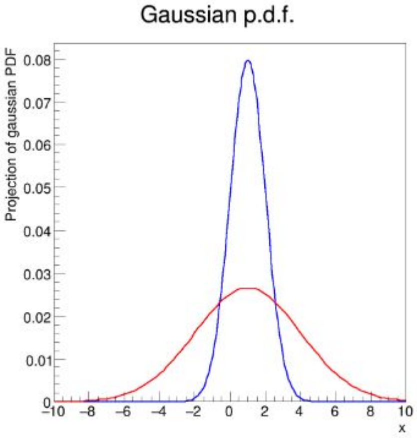


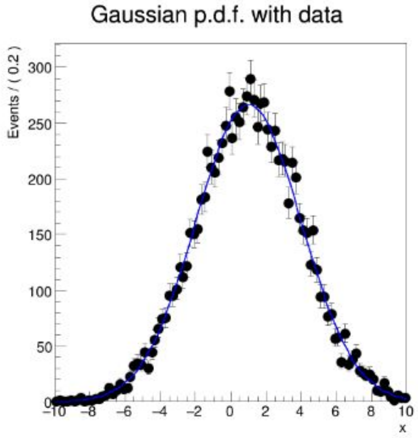


ROOT RooFit

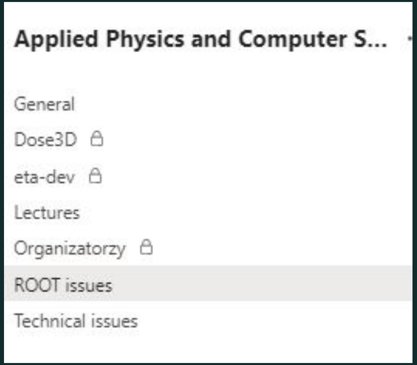
See tutorials, RooFit

- ➔ Toolkit for modeling the expected distribution of events in a physics analysis
- ➔ Models can be used to perform likelihood fits, produce plots, and generate "toy Monte Carlo" samples for various studies
- ➔ Extra data types
  `RooRealVar, RooDataHist, RooDataSet...`





The instructions show you the way, but it won't go for you...


Applied Physics and Computer S...

- General
- Dose3D 🔒
- eta-dev 🔒
- Lectures
- Organizatorzy 🔒
- ROOT issues
- Technical issues