

# Feedback from the Analysis Community

Jochen Klein

CERN

ALICE Offline week

Nov. 25th, 2015



# thanks for invitation to give feedback

in the following:

- ▶ some **points from analyzers' perspective**
  - ▶ you don't have to agree with them
  - ▶ but appreciated if they are acknowledged as a view
  - ▶ better proposals for improvement are more than welcome
- ▶ this is **not a representative survey**  
could be done but needs more preparation
- ▶ collection of
  - ▶ personal impressions from interactions with analyzers
  - ▶ comments contributed from analysis community

## build system

**many advantages**, some (easily avoidable) issues:

- ▶ faster build times
- ▶ known what is actually build
- ▶ reproducible environment allows local debugging  
not yet widely known, but helpful for advanced users
- ▶ automatic loading of dependencies very useful  
but has not reached all the trains
- ▶ some unannounced upgrades, e.g. fastjets,  
not all implications clear to everybody
- ▶ recently many tags deployed to cvmfs,  
not always clear which ones to use  
(add comments and/or usability flags?)

## coding

- ▶ coverity for AliPhysics?  
we don't have to be as strict as with AliRoot  
but could help spotting problems
- ▶ how strict do we want to be with warnings? here for AliPhysics,  
different requirements than AliRoot
- ▶ guidelines on allowed dependencies  
some unintended ones introduced in AliPhysics
- ▶ recommendations for documentation?
  - ▶ ROOT-style comments
  - ▶ doxygen
- ▶ coding guidelines?

## git commits

- ▶ current workflow:  
send patch by email to somebody with commit rights
- ▶ time critical because of tagging  
committer has to process in time  
(this is what is used for the analysis trains,  
a day of delay can hurt)
- ▶ additional load on committers  
at least (and often only) check compilation and commit
- ▶ automatic check of compilation?  
(in reach with new build system now?)
- ▶ do we need this for PWG<sub>xx</sub> directories? or
  - ▶ let everybody commit to e.g. PWG<sub>xx</sub>/UserTasks?
  - ▶ gitlab with pull requests?
  - ▶ something else?

# LEGO trains

- ▶ daily tags convenient for full train running
- ▶ also testing of trains depends on daily tags  
limits turn-around times for testing
- ▶ if we have builds from the commit testing:  
faster deployment possible at least for the train tests?
- ▶ provide LEGO train docker container  
for local testing in full setup?

## analysis framework

- ▶ no dedicated developer  
only limited development
- ▶ work for ROOT 6 migration?
- ▶ future of data structures, e.g. AODs or nanoAODs  
who takes care?  
(nanoAODs available but not widely used)
- ▶ usage of VAF?
  - ▶ how much is it used?
  - ▶ what are further plans?

(dedicated talk later)

## communication

- ▶ good atmosphere on the mailing list, helpful comments on the mailing list
- ▶ little information flow through PWG conveners at least for some pWGs
- ▶ role of user representative was recently discontinued communication could benefit from further discussion
  - ▶ discussion started amongst juniors
- ▶ unclear priorities who defines them actually?



## summary

- ▶ in preparation of this talk now:  
no severe complaints on offline performance,  
people appear to be mostly happy
  - ▶ communication could be improved
  - ▶ wishes from the offline group  
to the analysis community?
- 

# Thank you!

