Git Openness

Facing the reality…

Few points for discussion

“Open-source” in some MC codes

- MCNP/MCNPX (https://mcnp.lanl.gov/)
  - “Export-controlled code”, only binary code of public releases available on explicit request
- Penelope (http://www.oecd-nea.org/tools/abstract/detail/nea-1525)
  - Only public released code available and downloadable on explicit request
- Fluka (http://fluka.org/)
  - Only binary code of public releases available on explicit subscription
- EGS/EGSnrc (https://www.nrc-cnrc.gc.ca/eng/solutions/advisory/egsnrc_index.html)
  - Free source code on GitHub (released code); open to pull-requests

Potential reasons…

- All adopting an “outdated” development model?
- Too strict license constraints?
- Loss of “trust“ by companies/agencies contractors?
- Too complex software for the general public?
- “Sensitive” code?
- … else?

- Likely… none of these apply to Geant4 itself ?

The Geant4 “open-source” model

- The Geant4 source code is provided for free
  - No particular restrictions or subscriptions required
- Providing one public release **every year**
  - Code guaranteed to satisfy all available validation tests and Q/A controls
- Validated patches to releases as necessary (2-3 every year in average)
- One public preview release of the new year release

➢ World-wide available on GitHub (https://github.com/Geant4/geant4)

➢ Open to merge requests, for potential integration in the development and/or patches

- Providing one development release **every month**
  - Code guaranteed to pass system testing only;
  - Validation and Q/A applied a-posteriori; related fixes introduced a-posteriori

❖ Available for CERN experiments and users on CVMFS

Geant4: a mission critical tool

- Geant4 is a mission critical tool for many applications
  - HEP, nuclear physics, medical, space, homeland security, etc.
  - These communities rely on proper maintenance and validation
- The mission of the Geant4 Collaboration:
  - Improve, extend, document and maintain the code base in the toolkit
  - Provide the best validation possible of the physics
  - Guarantee backwards compatibility and stability as much as possible and document migration of user's code to new releases whenever necessary
  - Ensure contributions (also external) are properly integrated, credited and supported
- Geant4, as Monte-Carlo code, provides a ***physics*** software:
  - A "fix" in multiple-scattering is not local to just multiple-scattering or the specific application making use of it…
  - … it can impact ***all*** physics results
  - A "fix" for low energy applications may break uses in HEP…
  - Naïve pull-requests cannot work and may severely danger validation aspects

Geant4 validation effort

- Amount of physics validation is large, and has to be examined carefully
  - Not an easy task which would be a burden with un-careful requests…
- Only ”validated” code must be used for physics analyses
  - At arbitrary time **t**, the development tree must not be used !
  - Development releases are just code under construction…
    - A physics model, a field stepper, a biasing feature may be set as -default- just for testing
  - Making a physics paper on non-validated code would make no sense
    - Must find a way to prevent this if development tree is made public
    - How much effort would it require if such case happens ?

Credits to developers

- Geant4 developments are often subject to publications by developers themselves
  - The Geant4 Collaboration must guarantee proper credit to developers and protect the best possible way their intellectual rights
  - A public code repository may discourage developers to expose their ongoing developments, part of their own research…
  - …. and consequently loose benefit from the ongoing validation of their code done by the Collaboration as part of the natural development process
  - Forking privately?
    - Not merging to master implies doubling resources for testing/validation
    - Requires appropriate adapted infrastructure
    - A late integration of a new feature/module may completely spoil and delay a release

Last but not the least…

- Based on personal experience…
  - Most MC code developers are… shy! 
  - They don’t like to see every single semicolon committed to the code become posted in real-time on… FaceBook 
  - Possible (greatly undesired) reaction: keep the code in a private branch as long as possible, until development is judged complete
    - … again, spoiling release integration and testing

Geant4 public code on GitHub

- Available since June 2016 (https://github.com/Geant4/geant4)
- Current number of forked repositories: **63**
- Historical total number of pull requests from users: **1**

Discussion