

Parallel 4B

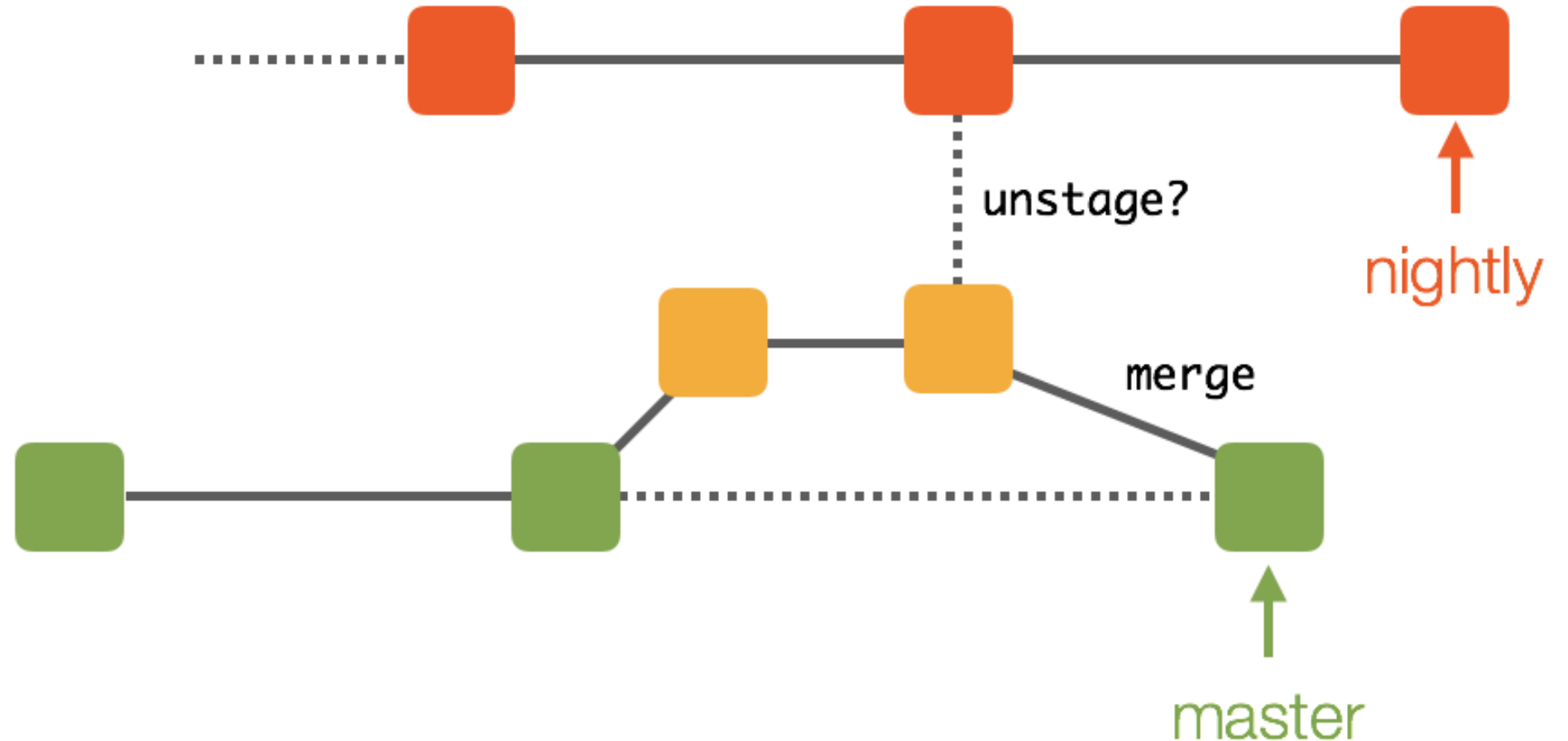
*Migration of Geant4 Version Control to Git
Working Session*

G.Cosmo, G.Folger, B.Morgan

Experiences from CMS & ATLAS

- CMS (*V.Ivantchenko*):
 - Using GitHub since end 2013; migrated from CVS; new development model well received by the collaboration and the developers
 - Rather centralised control of merge request; central role for release manager
 - Testing of a MR is started only after approval
 - Different active branches where developers can submit MR
 - Automation of testing process implemented through CMS specific "Bot" system
- ATLAS (*A.Dotti*):
 - Migrated recently from SVN to GitLab; positive feedback from developers since the beginning; much larger code base than Geant4...
 - Similar development model as the one proposed by our Task-Force, with some differences:
 - Developers use 'forking' of the repository to submit MR
 - Directly using Master trunk for testing MRs nightly builds (merged to the Master)
 - Important role of shifter in charge for review of MRs

Workflow proposed



- MR tested in isolation in Continuous testing; automatic 'almost' immediate response from testing
- MRs passing Continuous and code review (by shifter & coordinators) tested on 'stage' branch for nightlies
- Faulty MR unstaged and back to Continuous/review; others merged to Master (accepted)

Points of discussion addressed

- Generation of tags for 'accepted' merge requests
 - To be verified how this can be automated, based on MR description
- Testing cycle time for Continuous builds to be effective
 - Incremental builds and exploring ccache solution also adopted in Root
- Timescale and transition
 - To be done in 2018 during 'quiet' period (after a reference-tag or Beta release)
 - Switch to Git to be done as sharp cutoff with Master cloning last reference tag (Master always stable and working) and SVN repository switched to read-only mode
 - Proper documentation for developers to be provided before that with 'essential' Git commands (and examples) to use for implementing the process
- Addition of automatic formatting checks and Coverity analysis/reports
 - To be explored and eventually added in a later stage

Home Work

- Home work identified for current Task Force for the next months
 - Focused on identifying the tools/techniques to implement the process and try them out
 - Identify benefits of using simple Merge Requests vs. Forking
- Gradual implementation of different stages
 1. testing MR submission process and simple cycle of testing (Continuous/Nightly)
 2. code review process and MR iteration: exploit WIP features; MR documentation; automatic tagging
 3. tune build/testing process in Continuous build
 4. implement hooks to Bugzilla/JIRA
- Involve collaborators in try-out sessions on current Git prototype branch
 - Encourage participation in JIRA discussion thread

Thanks!

- JIRA discussion threads:
 - <https://jira-geant4.kek.jp/projects/GIT/issues/GIT-2?filter=allopenissues>
- Git try-out repository:
 - <https://gitlab.cern.ch/geant4/geant4-git-prototype>