OpenACS/EuroTcl 2022

GitLab CI pipelines for OpenACS development

Héctor Romojaro Gómez

Learn@WU Systemmanager

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

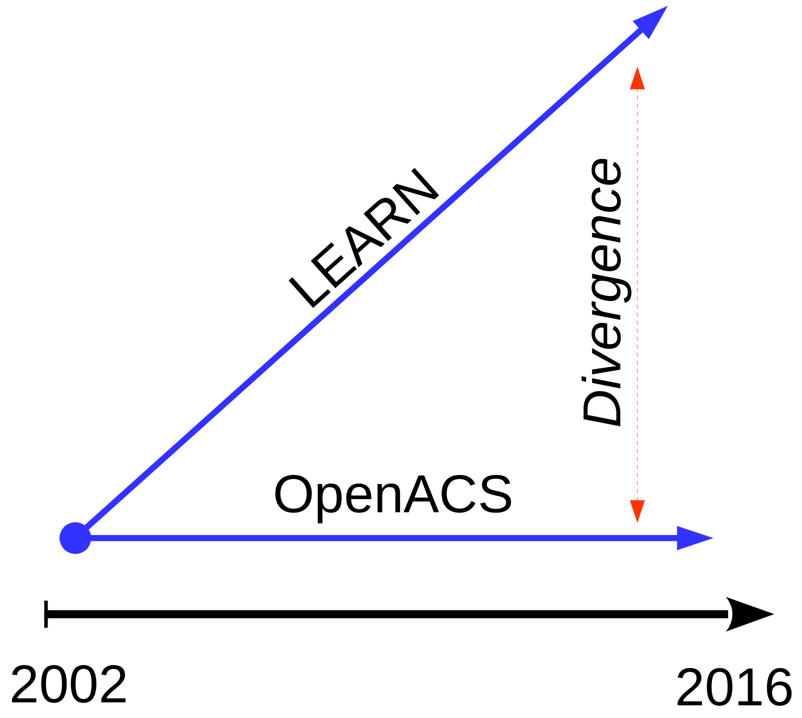
Learn@WU

- One of the world’s **most intensively used** E-learning platforms in higher education
- Based on OpenACS + NaviServer
- Started in 2002, designed for scalability

Some numbers:

- Up to 15 Mio hits and 3,3 Mio page impressions/day from registered users
- Up to 2500 concurrent users, over 250 views/sec
- Average response time on views less than 0.05 sec
- More than 120.000 learning resources have been developed since 2002
- Single instance

Back in 2016...

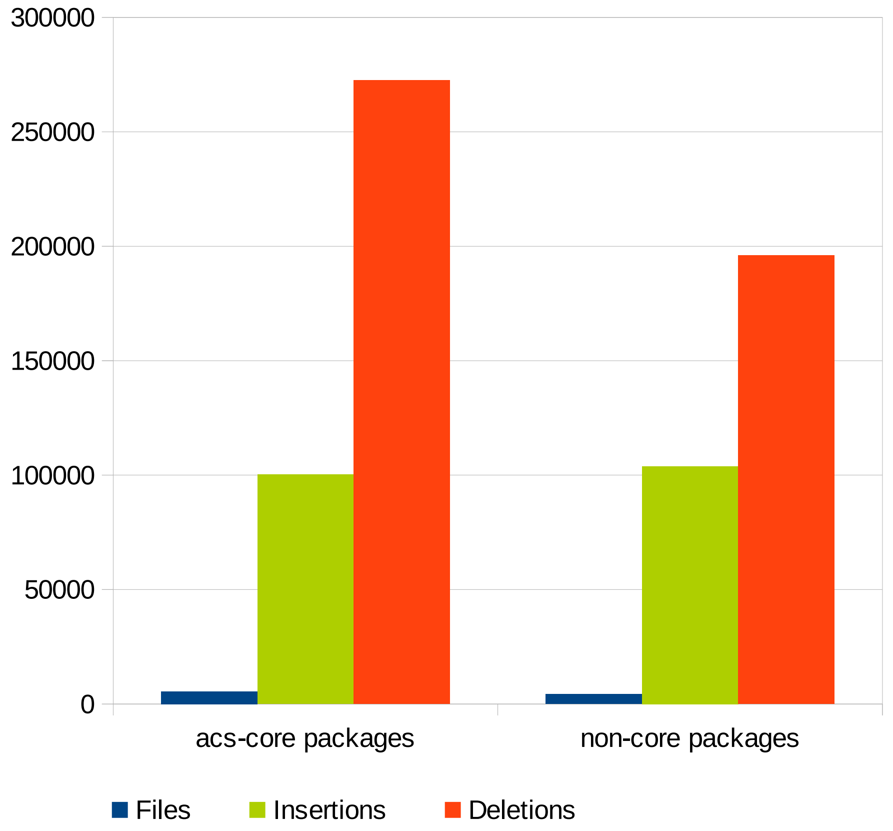


- No Upstream merging
- Package upgrades have been done occasionally
- Divergence increased over time
- It made more complex to integrate from the community
  - Bug Fixes
  - New features
  - Security fixes
  - Performance improvements
- Tests run manually

Numbers

Divergence size as of March 2016

- `git diff ...`
- Core packages
  - Files changed: 5514
  - Insertions: 100222
  - Deletions: 272529
- Non Core packages
  - Files changed: 4320
  - Insertions: 103791
  - Deletions: 196018
- Ignoring catalog files, ajaxhelper, white spaces and local packages.

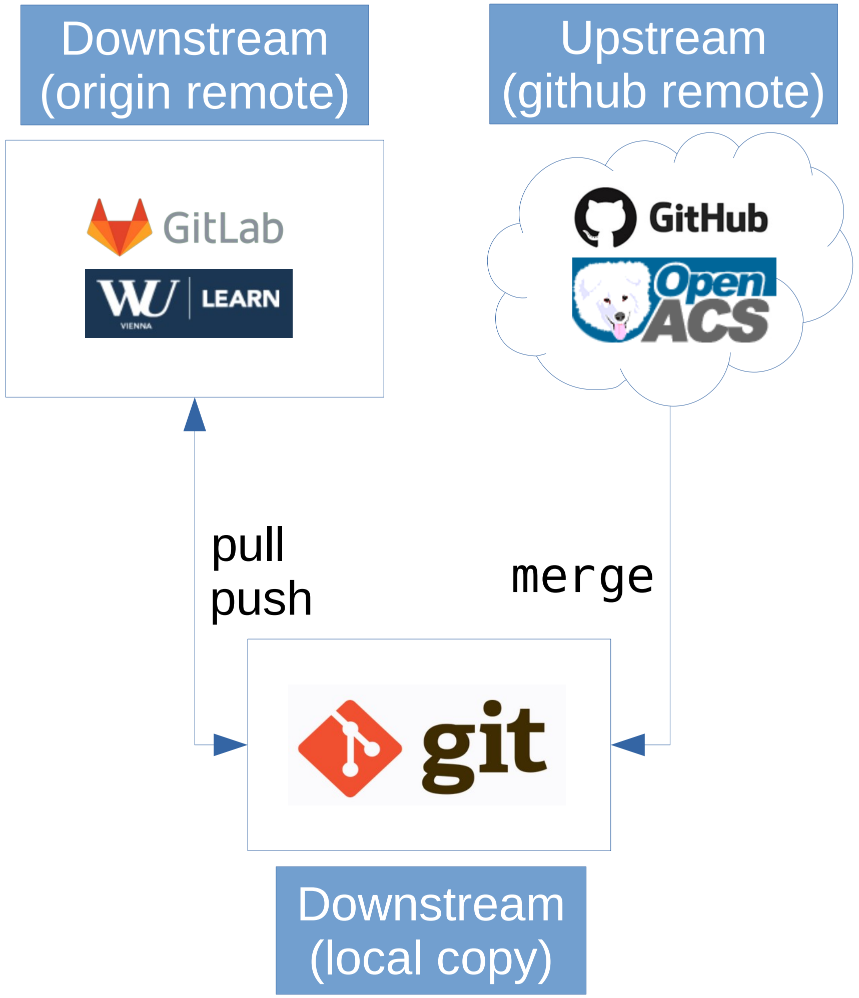


The Challenge

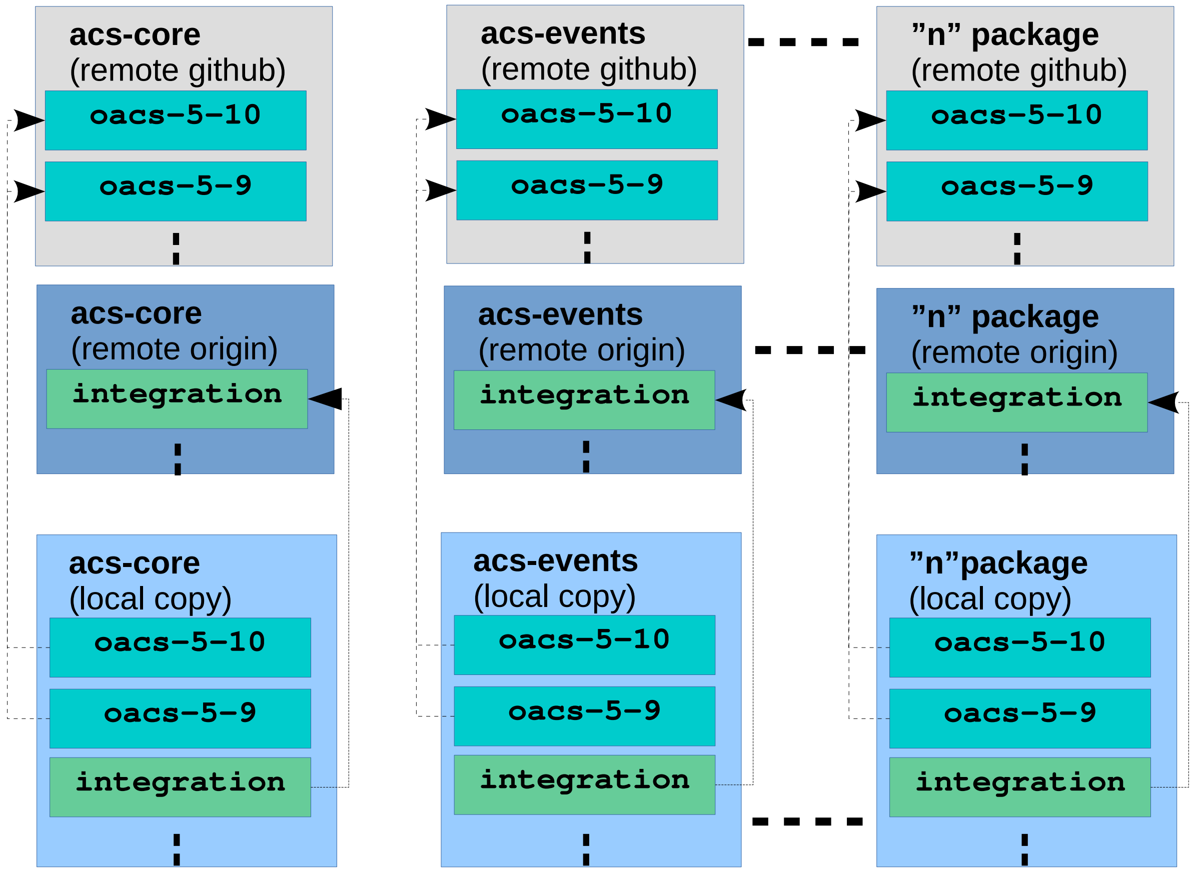
- Reduce divergency to the **minimum necessary**.
- Integrate OpenACS upstream code **efficiently**
  - Use Github’s OpenACS repository as a *remote* for our Git repository
  - Import new upstream commits quickly and easily
  - Merge OpenACS code with ours keeping the history of both
- Increase Software Quality
  - Decrease duplicity and redundancy
  - Trigger automated tests automatically

The Solution

- Use two Git remotes (*origin* for local code and *github* for upstream)
- Split *non-core* packages into separate repositories, matching upstream's Github structure
- Initial merge on common ancestor
  - Find the upstream branch more similar to downstream
- Subsequent merges are easy and fast
- *myrepos* to manage multiple repositories easily
- *GitLab* local instance for CI pipelines
  - Trigger automated tests on every new commit in the integration branch, including upstream merges.

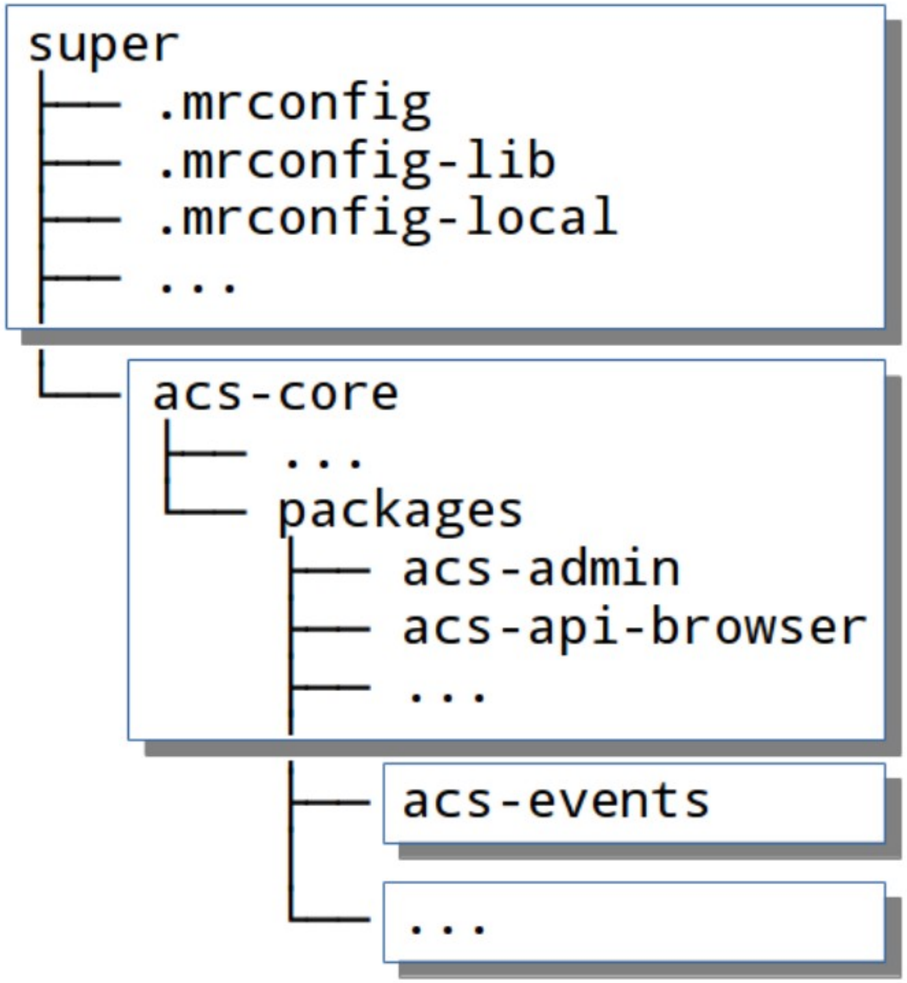


Integrated git structure



- 1 repository for *acs-core*
- "n" repositories for *non-core* packages
  - n = 66
- 2 remotes per repository
  - Github
  - Origin
- All branches available from local copy
  - `oacs-x-y` from github
  - `integration` from origin
- `oacs-x-y` is merged into `integration` daily
- All local development is in `integration`
- All history, local and upstream, is preserved

Directory structure

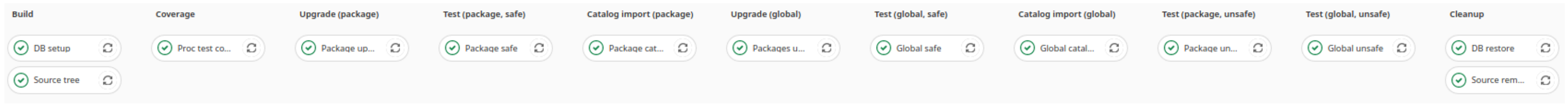


```
super
├── .mrconfig
├── .mrconfig-lib
├── .mrconfig-local
├── ...
└── acs-core
    ├── ...
    └── packages
        ├── acs-admin
        ├── acs-api-browser
        ├── ...
        ├── acs-events
        └── ...
```

- *Super-repository* stores the myrepos and common Gitlab CI config.
- *acs-core* repository contains OpenACS core and local packages with no upstream counterpart.
- *non-core* repositories, one per non-core package.

GitLab

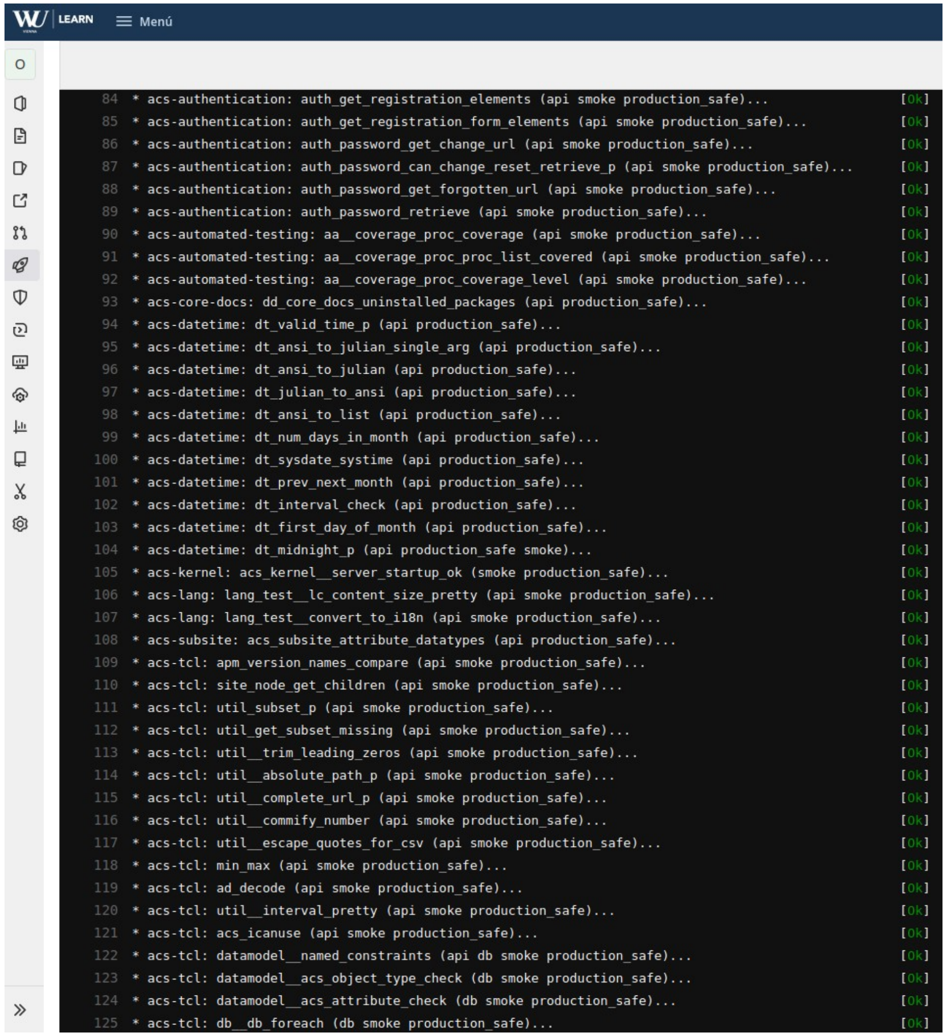
- Web-based DevOps lifecycle tool
- Git repository manager
- Integrated Web IDE
- Free software (Gitlab CE), with an open-core development model
- Private repositories, groups, forks, permissions, stats…
- On premises
- Store CI/CD config (`gitlab-ci.yml`) in *super-repository*
- Integrated CI/CD pipelines
  - Every change in the integration branch triggers the pipeline.
  - GitLab *runners* execute the jobs in docker containers with a running NaviServer
  - Jobs are run in stages



Build	Coverage	Upgrade (package)	Test (package, safe)	Catalog import (package)	Upgrade (global)	Test (global, safe)	Catalog import (global)	Test (package, unsafe)	Test (global, unsafe)	Cleanup
✓ DB setup	✓ Proc test co...	✓ Package up...	✓ Package safe	✓ Package cat...	✓ Packages u...	✓ Global safe	✓ Global catal...	✓ Package un...	✓ Global unsafe	✓ DB restore
✓ Source tree										✓ Source rem...

Job internals

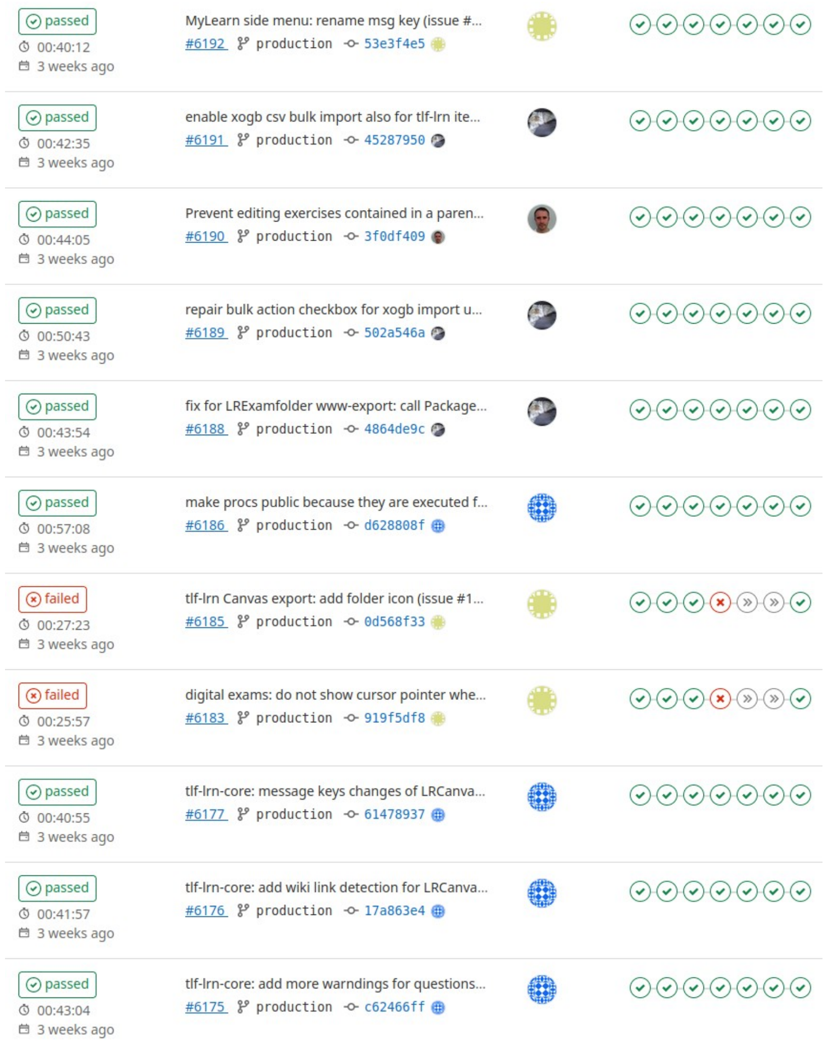
- Run by GitLab runners on docker containers with a running NaviServer
- Access to a shared PG instance, where the DB pool is
- ci-* scripts on */www/* accessed via cURL by GitLab
- Results output using *ns_write*
- Info retrieved via grep from server logs
  - Success of the job
  - `egrep -wq 'ci-tests:... '`
- Artifacts
  - error.log
  - rebuild_db



```
* acs-authentication: auth_get_registration_elements (api smoke production_safe)...                [Ok]
* acs-authentication: auth_get_registration_form_elements (api smoke production_safe)...           [Ok]
* acs-authentication: auth_password_get_change_url (api smoke production_safe)...                  [Ok]
* acs-authentication: auth_password_can_change_reset_retrieve_p (api smoke production_safe)...     [Ok]
* acs-authentication: auth_password_get_forgotten_url (api smoke production_safe)...               [Ok]
* acs-authentication: auth_password_retrieve (api smoke production_safe)...                        [Ok]
* acs-automated-testing: aa__coverage_proc_coverage (api smoke production_safe)...                 [Ok]
* acs-automated-testing: aa__coverage_proc_proc_list_covered (api smoke production_safe)...        [Ok]
* acs-automated-testing: aa__coverage_proc_coverage_level (api smoke production_safe)...           [Ok]
* acs-core-docs: dd_core_docs_uninstalled_packages (api production_safe)...                        [Ok]
* acs-datetime: dt_valid_time_p (api production_safe)...                                           [Ok]
* acs-datetime: dt_ansi_to_julian_single_arg (api production_safe)...                              [Ok]
* acs-datetime: dt_ansi_to_julian (api production_safe)...                                         [Ok]
* acs-datetime: dt_julian_to_ansi (api production_safe)...                                         [Ok]
* acs-datetime: dt_ansi_to_list (api production_safe)...                                           [Ok]
* acs-datetime: dt_num_days_in_month (api production_safe)...                                      [Ok]
* acs-datetime: dt_sysdate_systime (api production_safe)...                                        [Ok]
* acs-datetime: dt_prev_next_month (api production_safe)...                                        [Ok]
* acs-datetime: dt_interval_check (api production_safe)...                                         [Ok]
* acs-datetime: dt_first_day_of_month (api production_safe)...                                     [Ok]
* acs-datetime: dt_midnight_p (api production_safe smoke)...                                       [Ok]
* acs-kernel: acs_kernel__server_startup_ok (smoke production_safe)...                             [Ok]
* acs-lang: lang_test__lc_content_size_pretty (api smoke production_safe)...                       [Ok]
* acs-lang: lang_test__convert_to_i18n (api smoke production_safe)...                              [Ok]
* acs-subsite: acs_subsite_attribute_datatypes (api production_safe)...                            [Ok]
* acs-tcl: apm_version_names_compare (api smoke production_safe)...                                [Ok]
* acs-tcl: site_node_get_children (api smoke production_safe)...                                   [Ok]
* acs-tcl: util_subset_p (api smoke production_safe)...                                            [Ok]
* acs-tcl: util_get_subset_missing (api smoke production_safe)...                                  [Ok]
* acs-tcl: util__trim_leading_zeros (api smoke production_safe)...                                 [Ok]
* acs-tcl: util__absolute_path_p (api smoke production_safe)...                                    [Ok]
* acs-tcl: util__complete_url_p (api smoke production_safe)...                                     [Ok]
* acs-tcl: util__commify_number (api smoke production_safe)...                                     [Ok]
* acs-tcl: util__escape_quotes_for_csv (api smoke production_safe)...                              [Ok]
* acs-tcl: min_max (api smoke production_safe)...                                                  [Ok]
* acs-tcl: ad_decode (api smoke production_safe)...                                                [Ok]
* acs-tcl: util__interval_pretty (api smoke production_safe)...                                    [Ok]
* acs-tcl: acs_icanuse (api smoke production_safe)...                                              [Ok]
* acs-tcl: datamodel__named_constraints (api db smoke production_safe)...                          [Ok]
* acs-tcl: datamodel__acs_object_type_check (db smoke production_safe)...                          [Ok]
* acs-tcl: datamodel__acs_attribute_check (db smoke production_safe)...                            [Ok]
* acs-tcl: db__db_foreach (db smoke production_safe)...                                            [Ok]
```

Stages: Order

- Try to avoid DB rebuild
- Gitlab runners in docker containers execute jobs in stages:
  1. Build (DB and source tree setup)
  2. Test coverage
  3. Package upgrade
  4. Package safe tests (*production_safe*)
  5. Package message catalog import
  6. Global package upgrade
  7. Global safe tests (*production_safe*)
  8. Global message catalog import
  9. Package unsafe tests
  10. Global unsafe tests
  11. Cleanup (DB rebuild if tainted and source tree removal)



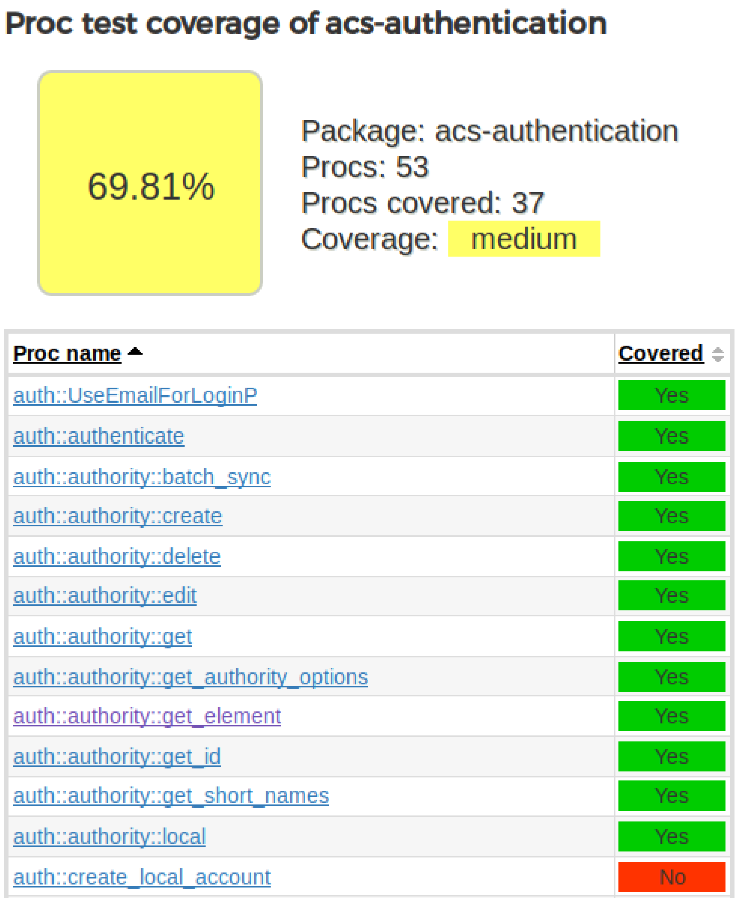
Status	Commit	Stages
passed 00:40:12 3 weeks ago	MyLearn side menu: rename msg key (issue #... #6192 production 53e3f4e5	✓ ✓ ✓ ✓ ✓ ✓ ✓
passed 00:42:35 3 weeks ago	enable xogb csv bulk import also for tlf-lrn ite... #6191 production 45287950	✓ ✓ ✓ ✓ ✓ ✓ ✓
passed 00:44:05 3 weeks ago	Prevent editing exercises contained in a paren... #6190 production 3f0df409	✓ ✓ ✓ ✓ ✓ ✓ ✓
passed 00:50:43 3 weeks ago	repair bulk action checkbox for xogb import u... #6189 production 502a546a	✓ ✓ ✓ ✓ ✓ ✓ ✓
passed 00:43:54 3 weeks ago	fix for LRExamfolder www-export: call Package... #6188 production 4864de9c	✓ ✓ ✓ ✓ ✓ ✓ ✓
passed 00:57:08 3 weeks ago	make procs public because they are executed f... #6186 production d628808f	✓ ✓ ✓ ✓ ✓ ✓ ✓
failed 00:27:23 3 weeks ago	tlf-lrn Canvas export: add folder icon (issue #1... #6185 production 0d568f33	✓ ✓ ✓ ✗ » » ✓
failed 00:25:57 3 weeks ago	digital exams: do not show cursor pointer whe... #6183 production 919f5df8	✓ ✓ ✓ ✗ » » ✓
passed 00:40:55 3 weeks ago	tlf-lrn-core: message keys changes of LRCanva... #6177 production 61478937	✓ ✓ ✓ ✓ ✓ ✓ ✓
passed 00:41:57 3 weeks ago	tlf-lrn-core: add wiki link detection for LRCanva... #6176 production 17a863e4	✓ ✓ ✓ ✓ ✓ ✓ ✓
passed 00:43:04 3 weeks ago	tlf-lrn-core: add more warndings for questions... #6175 production c62466ff	✓ ✓ ✓ ✓ ✓ ✓ ✓

Stages: Build

- Source tree retrieval using myrepos
- Database reservation
  - Thinned out production database (~270G)
  - Picked from a pool of databases
    - *gitlab-pipeline-free-** renamed to *gitlab-pipeline-$PIPELINE_ID*
  - Manage concurrency with *db-mutex* runner
    - Only one runner is tagged db-mutex
    - *db-mutex* is the only runner allowed to reserve/regenerate DB

Stages: Coverage

- Public procs covered by automated tests
- Fail if coverage decreases (enforces policy)
- New on 5.10
  - *aa::coverage::**
  - /test/admin/proc-coverage
  - Global and per package



Proc test coverage of acs-authentication

69.81%

Package: acs-authentication
Procs: 53
Procs covered: 37
Coverage: medium

Proc name	Covered
auth::UseEmailForLoginP	Yes
auth::authenticate	Yes
auth::authority::batch_sync	Yes
auth::authority::create	Yes
auth::authority::delete	Yes
auth::authority::edit	Yes
auth::authority::get	Yes
auth::authority::get_authority_options	Yes
auth::authority::get_element	Yes
auth::authority::get_id	Yes
auth::authority::get_short_names	Yes
auth::authority::local	Yes
auth::create_local_account	No

Stages: Upgrade

- Upgrades a single or all possible OpenACS packages
- Fail if dependency error or unsuccessful upgrade
- *Taints* DB if upgrade is performed, triggering a rebuild from the template on the Cleanup stage
- APM api
  - *apm_package_**
  - *apm_scan_packages*

Stages: Message catalog import

- Imports the message catalog files of a single or all possible packages.
- Detects conflicts.
- Detects changes on message keys (add, delete, update).
- Fail if changes or conflicts are detected without a package upgrade (enforces policy)
- *Taints* DB if changes are performed, triggering a rebuild from the template on the Cleanup stage.
- acs-lang api
  - *lang::catalog::import*
  - *lang::message::conflict_count*

Stages: Tests

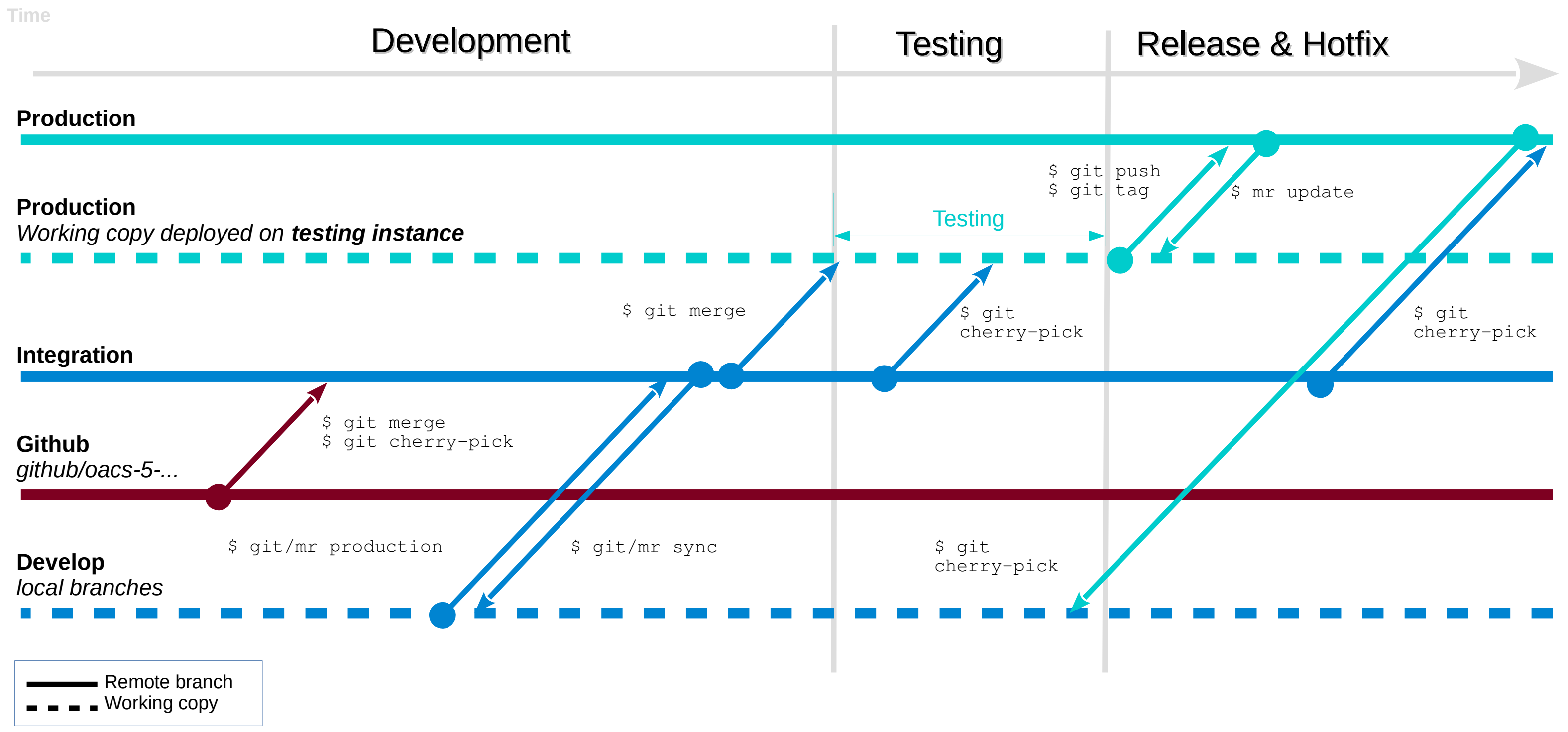
- Runs tests from *acs-automated-testing* on a single or all possible packages
- Fail if any test fails
- *Taints* DB if non *production_safe* tests are run
- Run as test user
  - *acs::test::user::**
- AA api
  - *aa_runseries*
  - *nsv_get aa_test cases*

Stages: Cleanup

- Deletes the source tree
- Restores or rebuilds DB
    - If DB is not *tainted*, just put it back in the pool
        - Rename *`gitlab-pipeline-$PIPELINE_ID`* to *`gitlab-pipeline-free-*`*
    - If DB is *tainted*, recreate it from the template DB
    - Manage concurrency with the *db-mutex* runner

...and the present!

- All work is commited to the integration branch
- Latest upstream code (5.10 branch) is merged into the integration branch daily
- Every push to the integration branch triggers the pipeline
- Internal releases every two weeks



Thanks for watching!

Some unnecessary links

- LEARN: https://learn.wu.ac.at/
- OpenACS: https://openacs.org
- OpenACS on Github: https://github.com/openacs
- NaviServer: http://naviserver.sourceforge.io/:
- myrepos: https://myrepos.branchable.com/
- GitLab: https://about.gitlab.com/