Lustre Code Quality
Lustre quality dashboard

WHY?

It's becoming increasingly important that we keep the community - and customers - updated on the health outlook of Lustre

- visibility of current/past quality
- risks
- definition of metrics
- one place to look
- gadgets to look from different angles
- feedback to development with code and process changes
Lustre quality dashboard

many possible metrics

- defects by testing
- defects in field
- defects retrospective - categorize defects
- parameters: number of users, users configuration, scale
- code analyzers
- commits/defects metrics
- code churn, layer, functions
- different metrics for development - based on testing results, reviews, analyze tools and etc
Lustre quality dashboard gadgets

- Jira based
  - internal bugs
  - field bugs
  - field impact (right bug severity?)
- Git based
  - number of changes
  - code churn
- Test based
  - test bugs
  - number of tests run on number of configs
- Analytic tools
  - code coverage
  - cppcheck, sparse and other code analysis tools
Lustre quality process

- regular
- review dashboard and metrics
- adjust metrics and other process
- act - make changes to other processes:
  - future development plans changes (improvements)
  - delivery process changes
  - etc
- and again
Lustre delivery

- big release cycles increase feedback
- improvements for quality take too much time to deliver

- rolling releases
- fast releases requires full CI and automation
Lustre Client

Distribution issue

- kabi
  - imported functions (d_mountpoint, rwlock_init)
- many customers - many different kernels
- many test and build configurations
- not all customers could build client
- klnd drivers separation and better LNET abstraction
- could melanox handle klnd?
Lustre analyzers

- AWS based Jenkins for OpenSFS
- vm based tests on AWS
- code coverage
- stat analyse - cppcheck

```
result = osd_oi_delete(osd_oti_get(env), osd, fid, oh->ot_handle, "--- Variable 'result' is assigned a value that OI_CHECK_FLD\d");
```
Lustre test result DB

The main idea is to speed up and facilitate the work of testers and eliminate errors caused by human factor.

- easy search
- autovetting
- regressions point
- restart testing, restart in a loop
- reports
  - user notification (Jira, Gerrit)
  - statistic (test run reports, release reports)
- follow automation (integration with different tools)