Removing Technical Debt

Ben Evans, Cray
LUG 2016
Defining Technical Debt

- Work that needs to be done to keep a codebase up to date
  - Removal of old code
  - Refactoring of existing code
  - Commenting and documenting code
  - Migrating to new standards (tabathon)
  - Naming things
  - Better grouping of related items
  - etc.
Avoiding creating technical debt

- Document what you wrote or changed
- Name things appropriately
- Don’t write ‘clever’ code
- Remember your audience
  - You
  - 6 months from now
  - On a weekend
  - At 3 AM
Coding Guidelines

- [http://wiki.lustre.org/Lustre_Coding_Style_Guidelines](http://wiki.lustre.org/Lustre_Coding_Style_Guidelines)
  - “Do what it says on the tin”
  - Naming:
    - Keep them distinct: rq_reqmsg and rq_repmsg
    - Use descriptive names: make_the_code_clearer, but do_not_let_things_get_out_of_hand
  - Commenting
    - Describe what a function does, input and outputs
Identifying Debt

- **Simple example**

```c
rc = mdd_changelog_on(env, mdd, 1);
rc = mdd_changelog_on(env, mdd, 0);
static int
mdd_changelog_on(const struct lu_env *env, struct mdd_device *mdd, int on)
{
    int rc = 0;

    if ((on == 1) && ((mdd->mdd_cl.mc_flags & CLM_ON) == 0)) {
        /*Code to enable changelog here*/
    } else if ((on == 0) && ((mdd->mdd_cl.mc_flags & CLM_ON) == CLM_ON)) {
        /* Code to disable changelog here */
    }
    return rc;
}
```
Cleaning it up

```c
rc = mdd_changelog_on(env, mdd);
rc = mdd_changelog_off(env, mdd);

static int mdd_changelog_on(const struct lu_env *env,
struct mdd_device *mdd)
{
/* Code to enable changelog here */
}
static int mdd_changelog_off(const struct lu_env
*env, struct mdd_device *mdd)
/* Code to disable changelog here */
}```
Things left out of example

- Adding comments
- Renaming variables
- Testing
- Reviews
- …
What this means to users

● Fun, quirky behavior!
  ● Register/unregister cl users on an MDS
  ● View and purge records from a client
  ● View currently registered users (and other info) using cat

● Things labeled (DANGEROUS)
  ● Get rid of the options labeled “Obsolete”
  ● Move “testing” to it’s own command

● multiple options that do the same thing
  ● lctl –ignore_errors vs. ignore_errors
Questions?