



Features based development workflow

Antonio De Marco
antonio@nuvole.org

Andrea Pescetti
andrea@nuvole.org

How can multiple developers work
together on the same project?

Real life example



Peter

Location: Barcelona, Spain

Plans for Friday evening: working on stuff.



Antonio

Location: Brussels, Belgium

Plans for Friday evening: couple of Belgian beers, he'll work later.

1. Sharing database dumps



Peter

```
$ svn up  
U      dump.sql  
Updated to revision 33.
```

```
..working...  
..working...
```

```
$ drush sql-dump --result-file="dump.sql"  
$ svn ci -m "Work done."
```

Later that night...



Antonio

..working...

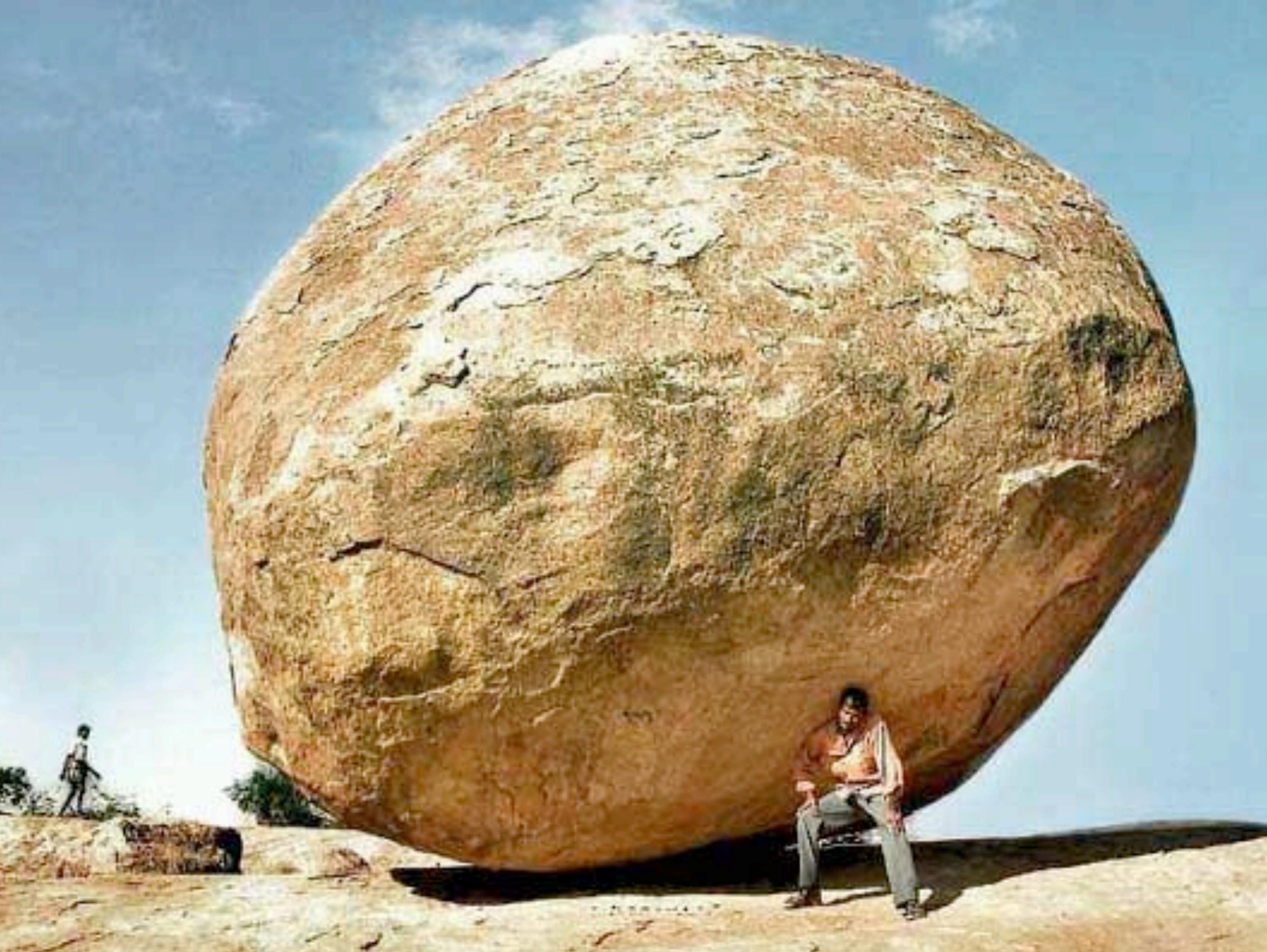
..working...

```
$ drush sql-dump --result-file="dump.sql"  
$ svn ci -m "Work done. Loves Leffe."
```

Peter's work is lost!

Major drawbacks

- Not ideal for a distributed team
- Makes it difficult to push settings to production
- Content and settings are mixed in one db dump
- Easy to lose control



2. Put everything in code

Major benefits

- Code can be versioned
- Conflicts can be solved
- Content and settings are separated
- Easy to push changes to production

Features: storing configuration in code

EDIT COMPONENTS:**Strongarm**

- ant_advocacy
- ant_organization
- ant_pattern_advocacy
- ant_pattern_organization
- ant_php_advocacy
- ant_php_organization
- cache
- cck_fieldgroup_tabs_description_organization

CCK

news-field_news_image

Contexts

feature_news_list

CTools export API

context:context:3 strongarm:strongarm:1

Imagecache presets

news-badge news-m news-s

Content types

news

Permissions

- create news content
- delete any news content
- delete own news content
- edit any news content
- edit own news content

Strongarm

- comment_anonymous_news comment_controls_news
- comment_default_mode_news comment_default_order_news
- comment_default_per_page_news comment_form_location_news
- comment_news comment_preview_news comment_subject_field_news
- content_complete_news content_profile_use_news node_options_news
- pathauto_node_news_pattern upload_news

Views

feature_news

Views API

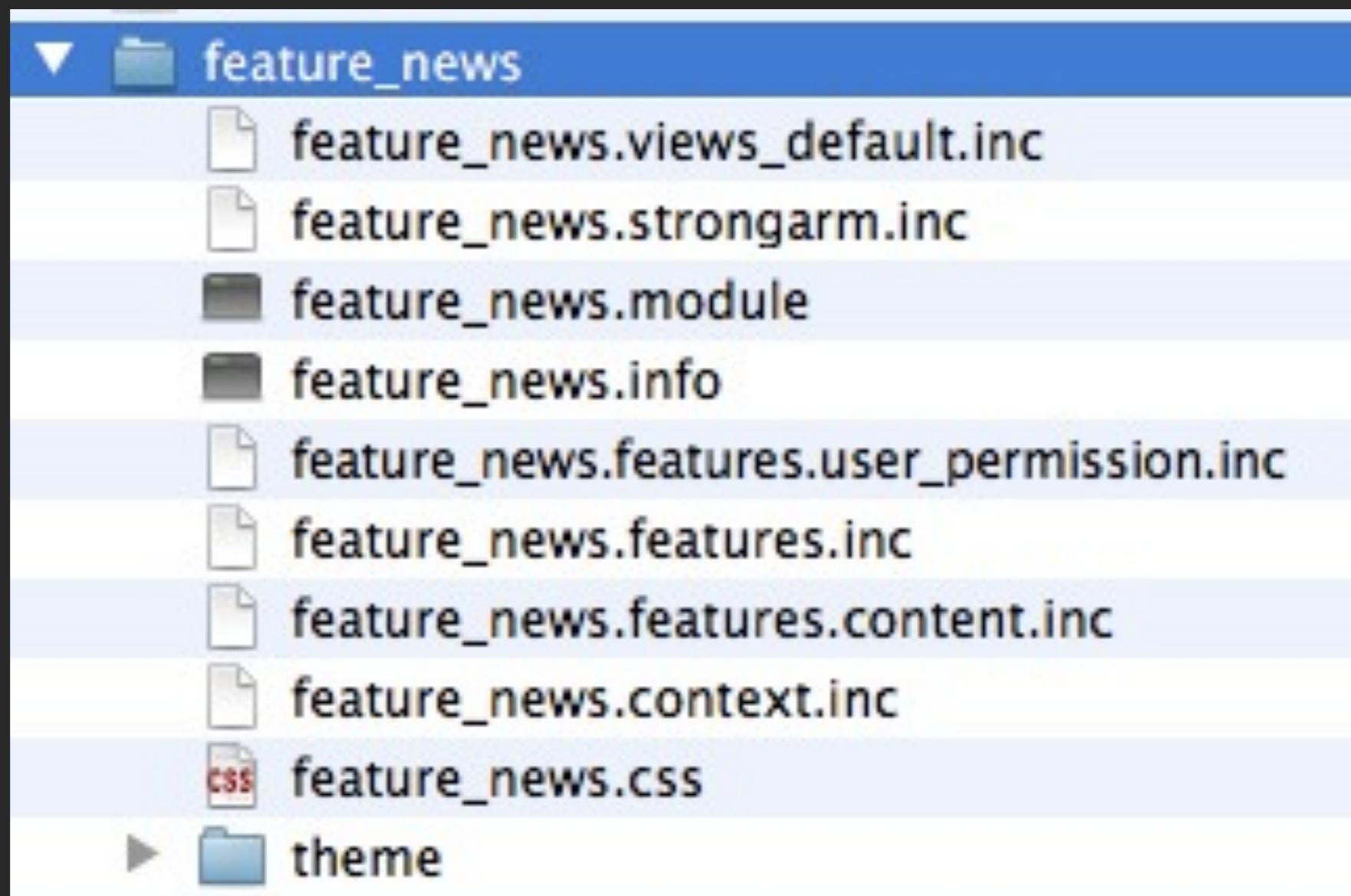
api:2

Dependencies

- context
- features
- imagecache
- imagefield
- strongarm
- views

- Normal
- Auto-detected
- Provided by dependency

Download feature



feature_news.info

```
core = "6.x"
dependencies[] = "context"
dependencies[] = "features"
dependencies[] = "imagecache"
dependencies[] = "imagefield"
description = "Publish news."
features[content][] = "news-field_news_image"
features[context][] = "feature_news_list"
features[ctools][] = "context:context:3"
features[ctools][] = "strongarm:strongarm:1"
features[imagecache][] = "news-badge"
features[imagecache][] = "news-m"
features[imagecache][] = "news-s"
features[node][] = "news"
...
name = "Feature News"
package = "Features"
project = "feature_news"
version = "6.x-1.0-dev"
```

From DB to code and vice versa

```
$ # Dump your changes into code.  
$ drush features-update feature_name  
  
$ # Restore your changes into the db.  
$ drush features-revert feature_name
```

Want to add a component to a feature?

1. add it to the .info file
2. \$ drush features-update feature_name

Want to remove a component from a feature?

1. remove it from the .info file
2. \$ drush features-update feature_name

Features are modules.
Modules can be updated.

Project bootstrap

project.make
project.profile

project.make

```
core = "6.x"

; Contrib
projects[features][version] = "1.0"
projects[features][subdir] = "contrib"

projects[views][version] = "2.11"
projects[views][subdir] = "contrib"

projects[cck][version] = "2.8"
projects[cck][subdir] = "contrib"

...
```

project.profile

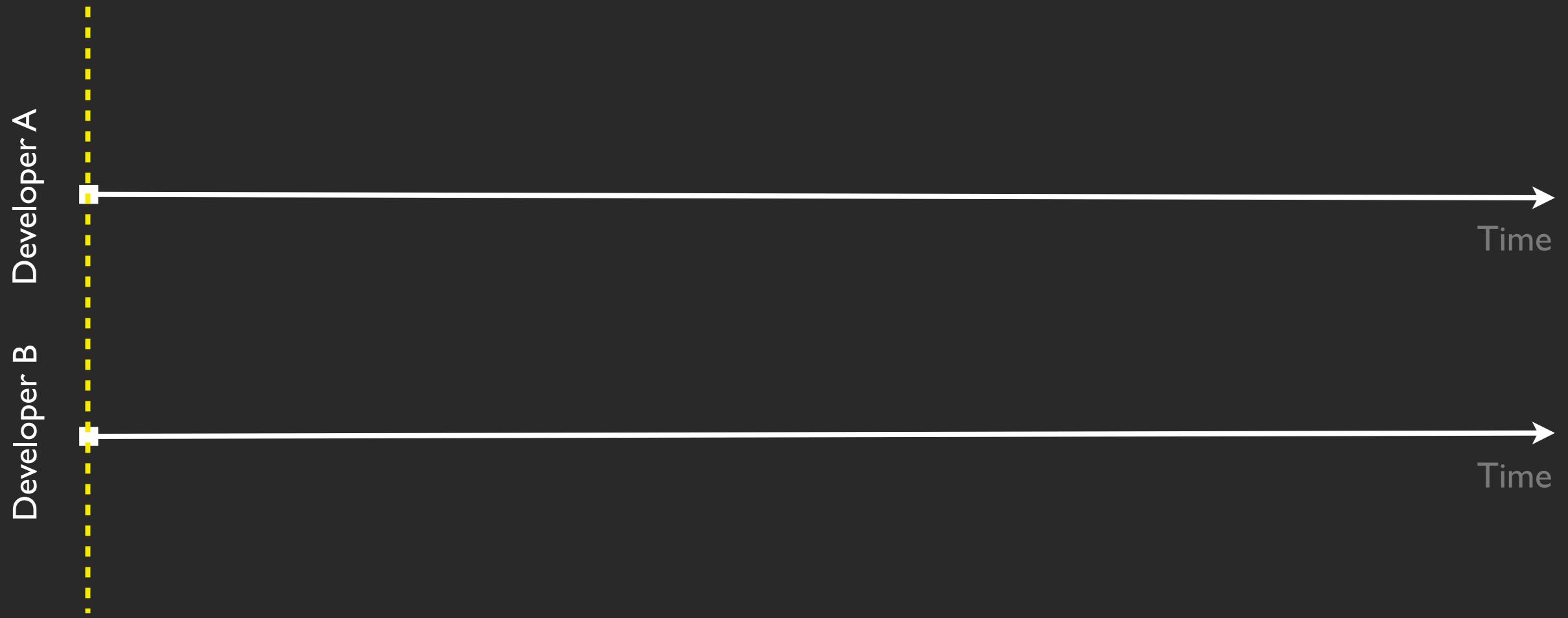
```
/**  
 * Implementation of hook_profile_details().  
 */  
function project_profile_details() {  
    return array(  
        'name' => 'Project',  
        'description' => 'Project Description',  
    );  
}  
  
/**  
 * Implementation of hook_profile_modules().  
 */  
function project_profile_modules() {  
  
    $modules = array(  
        'cck',  
        'views',  
        'features',  
        'feature_controller',  
        ...  
    );  
}
```



The Controller Feature

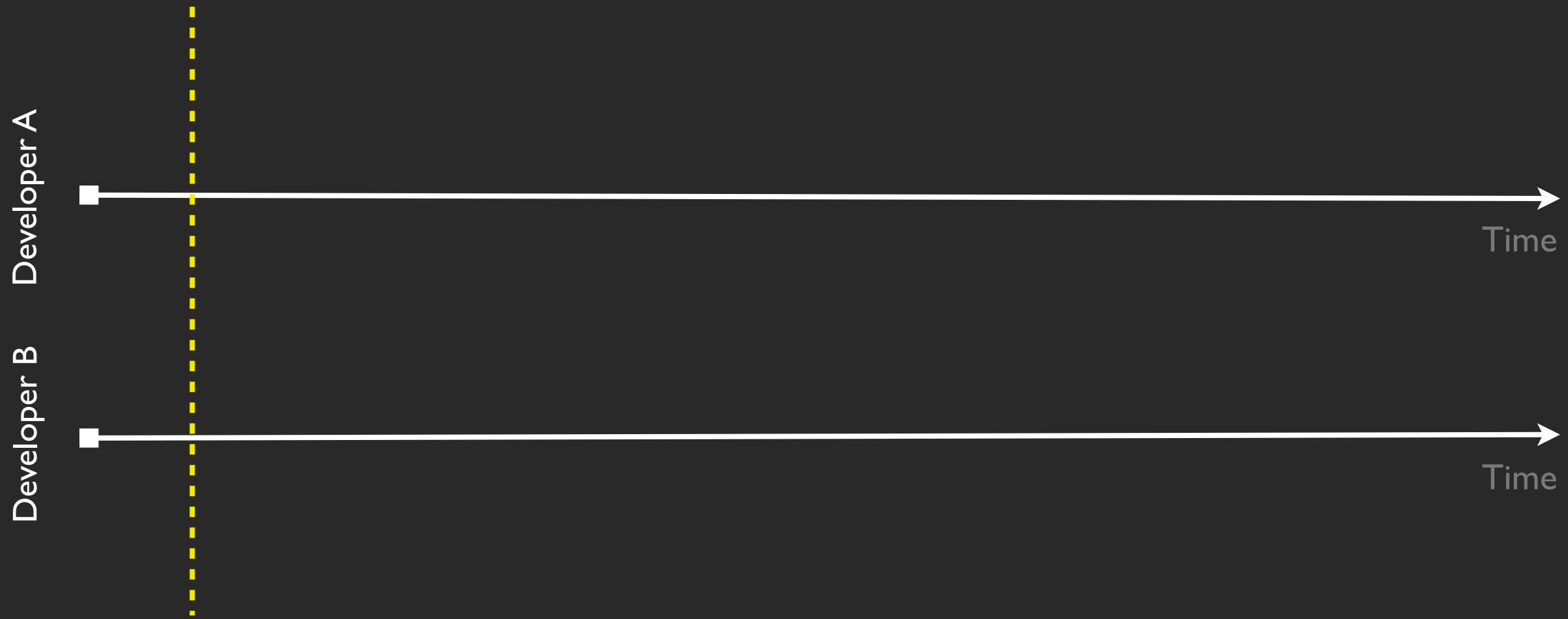
a.k.a. a feature to rule them all

Controller Feature Workflow



Start: both Developers run installation, Controller Feature is enabled.

Controller Feature Workflow

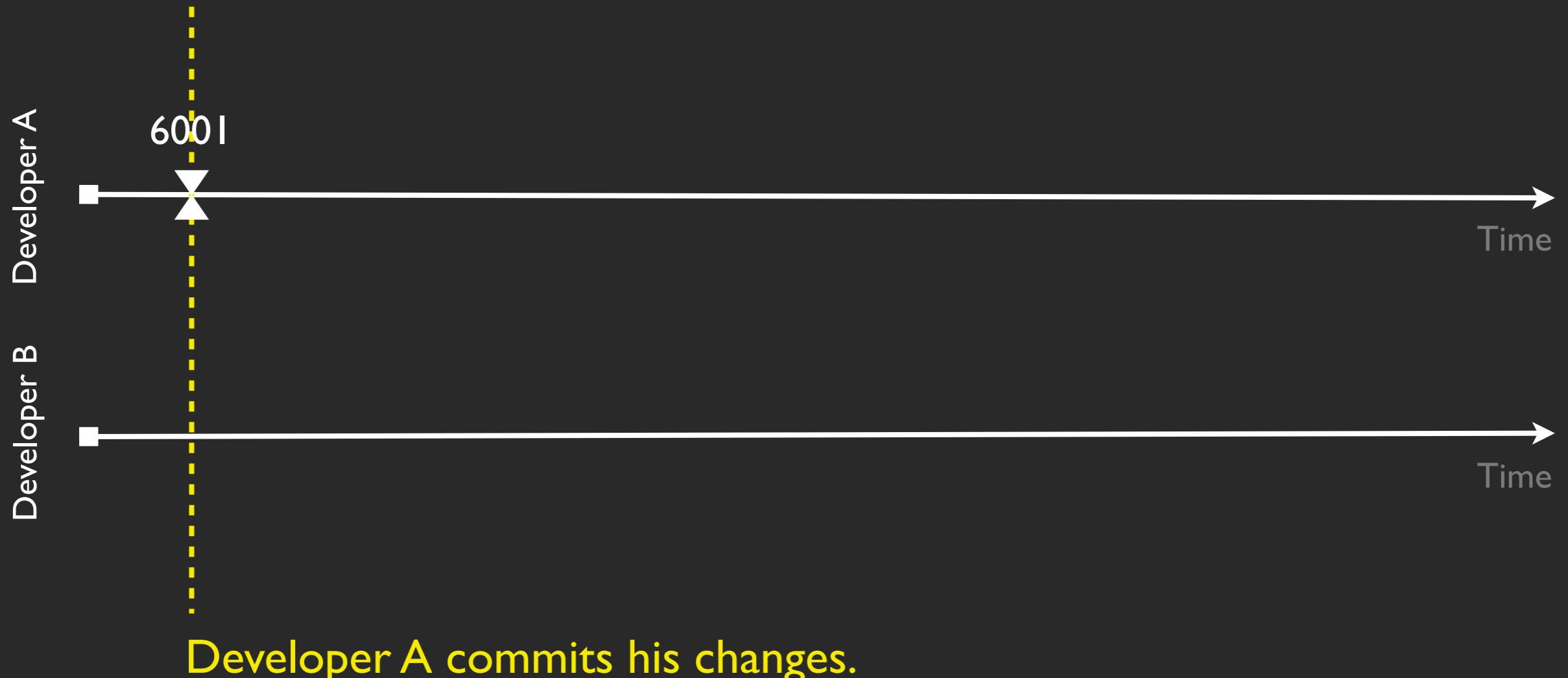


Developer A enables Rules module,
Developer B does other work on the project.

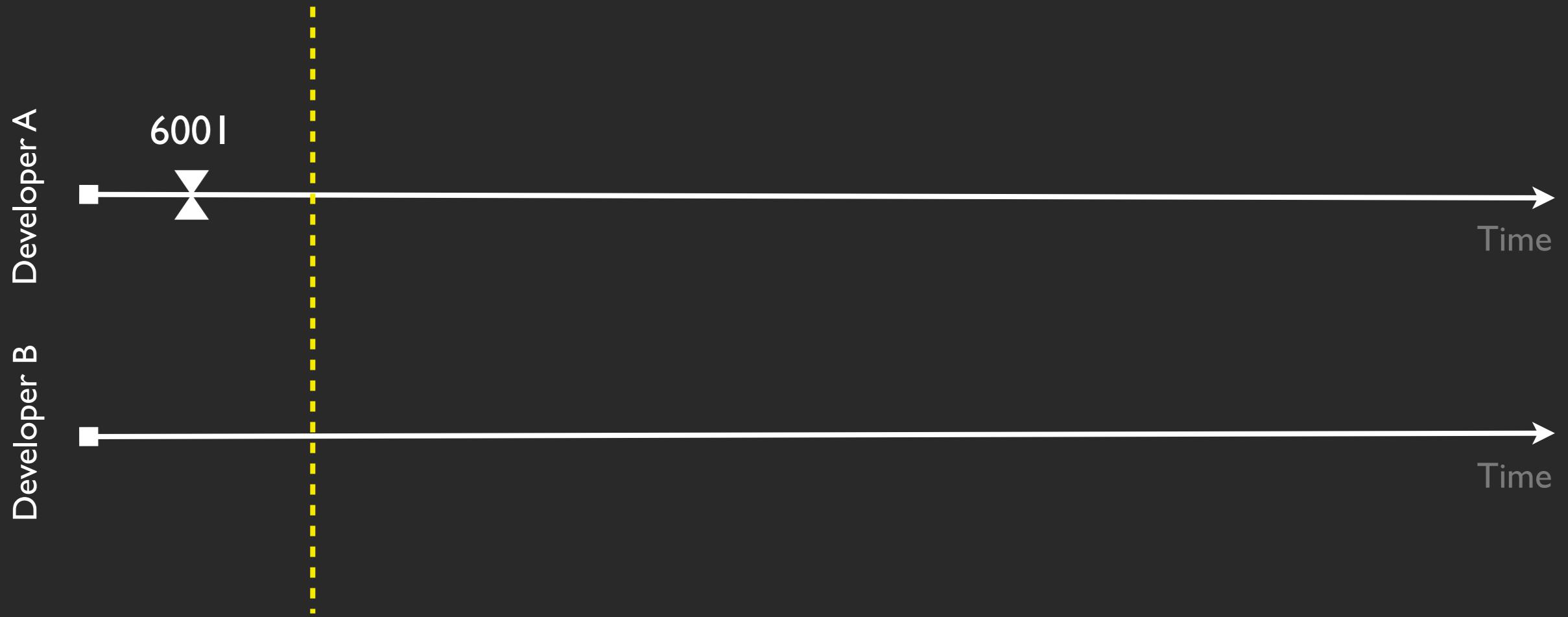
hook_update_N()

```
/**
 * Enabling Rules module
 */
function feature_controller_update_6001() {
  $return = array();
  $modules = array('rules');
  drupal_install_modules($modules);
  $return[] = array('success' => TRUE, 'query' => 'Enabling Rules module');
  return $return;
}
```

Controller Feature Workflow



Controller Feature Workflow

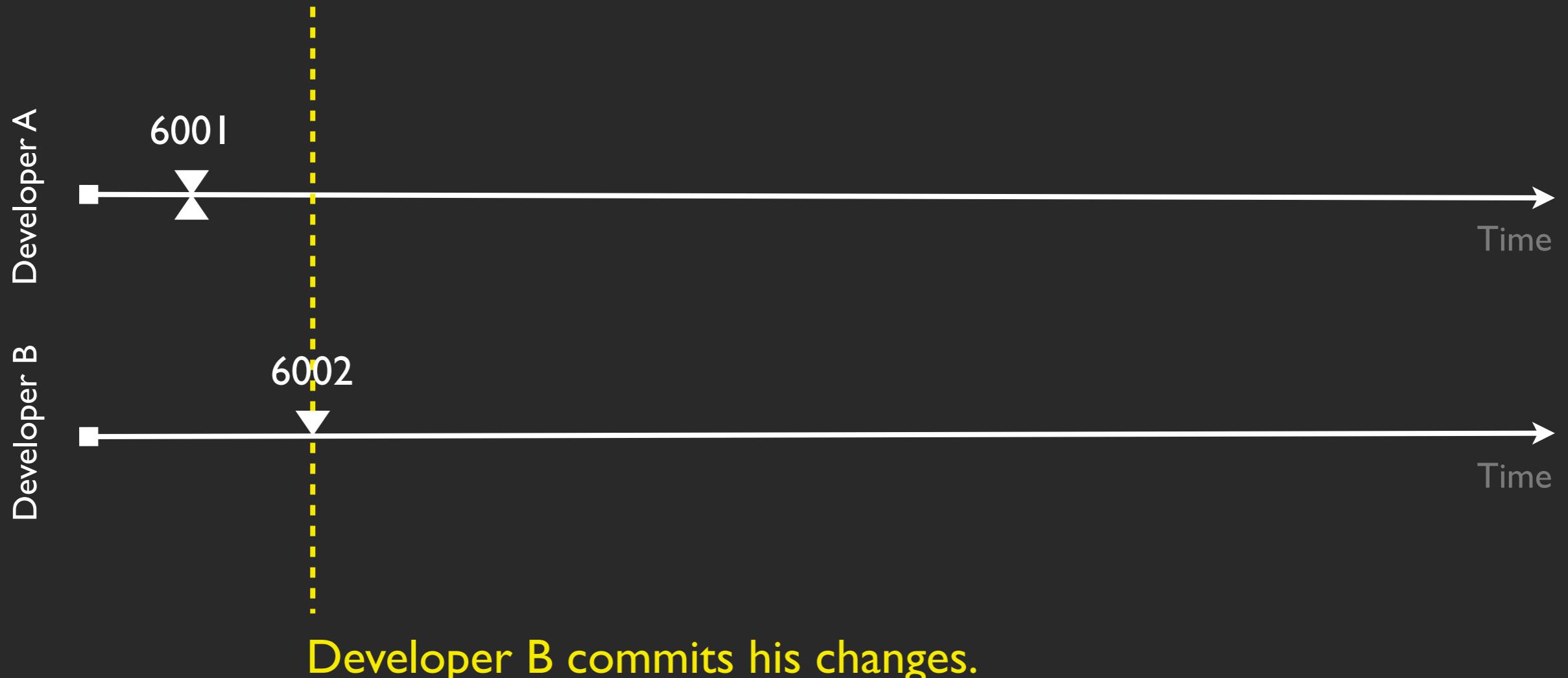


Developer B updates his local copy
and removes a role.

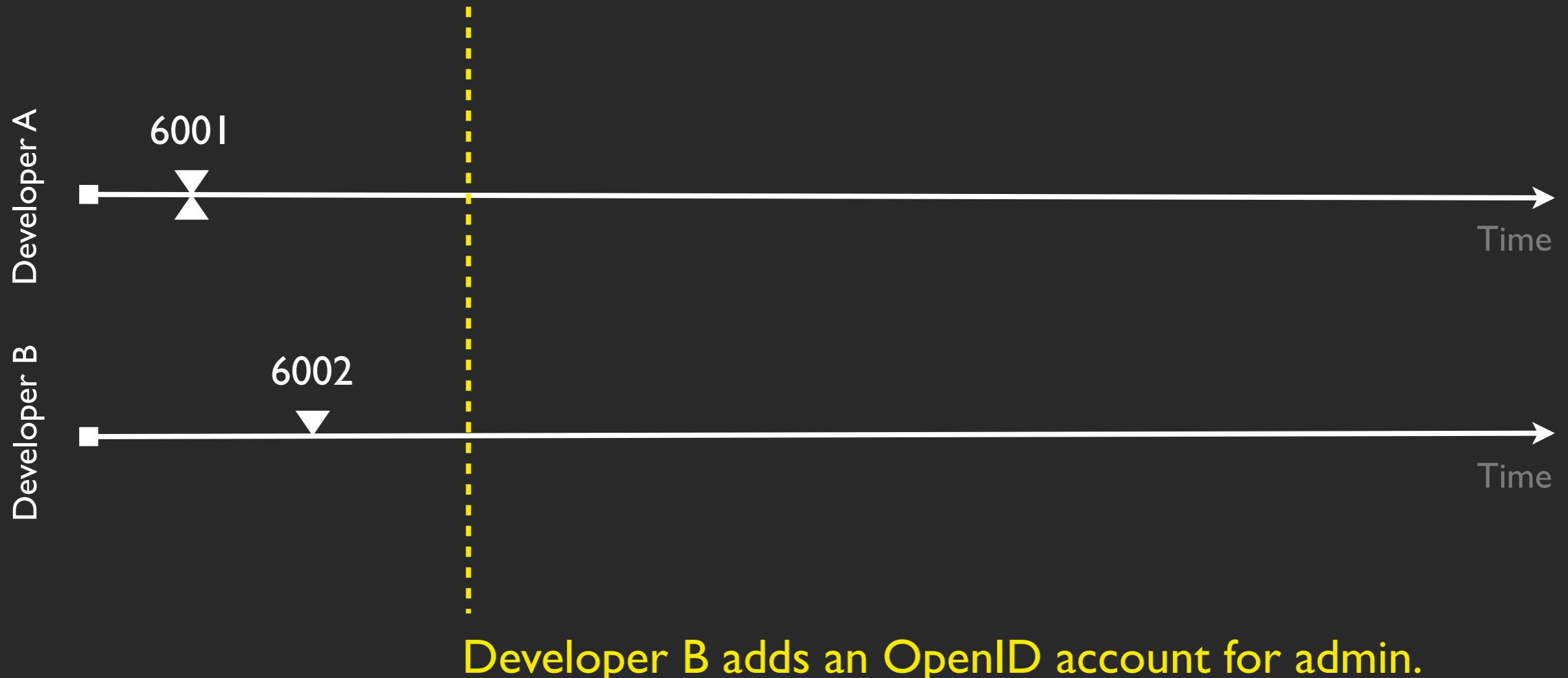
hook_update_N()

```
/**  
 * Removing contributor role  
 */  
function feature_controller_update_6002() {  
    $return = array();  
    $role_name = 'contributor';  
    $result = db_query("SELECT rid FROM {role} WHERE name='%s'", $role_name);  
    while ($role = db_fetch_object($result)) {  
        $rid = $role->rid;  
        $return[] = update_sql("DELETE FROM {role} WHERE rid = '$rid'");  
        $return[] = update_sql("DELETE FROM {users_roles} WHERE rid = '$rid'");  
    }  
    return $return;  
}
```

Controller Feature Workflow



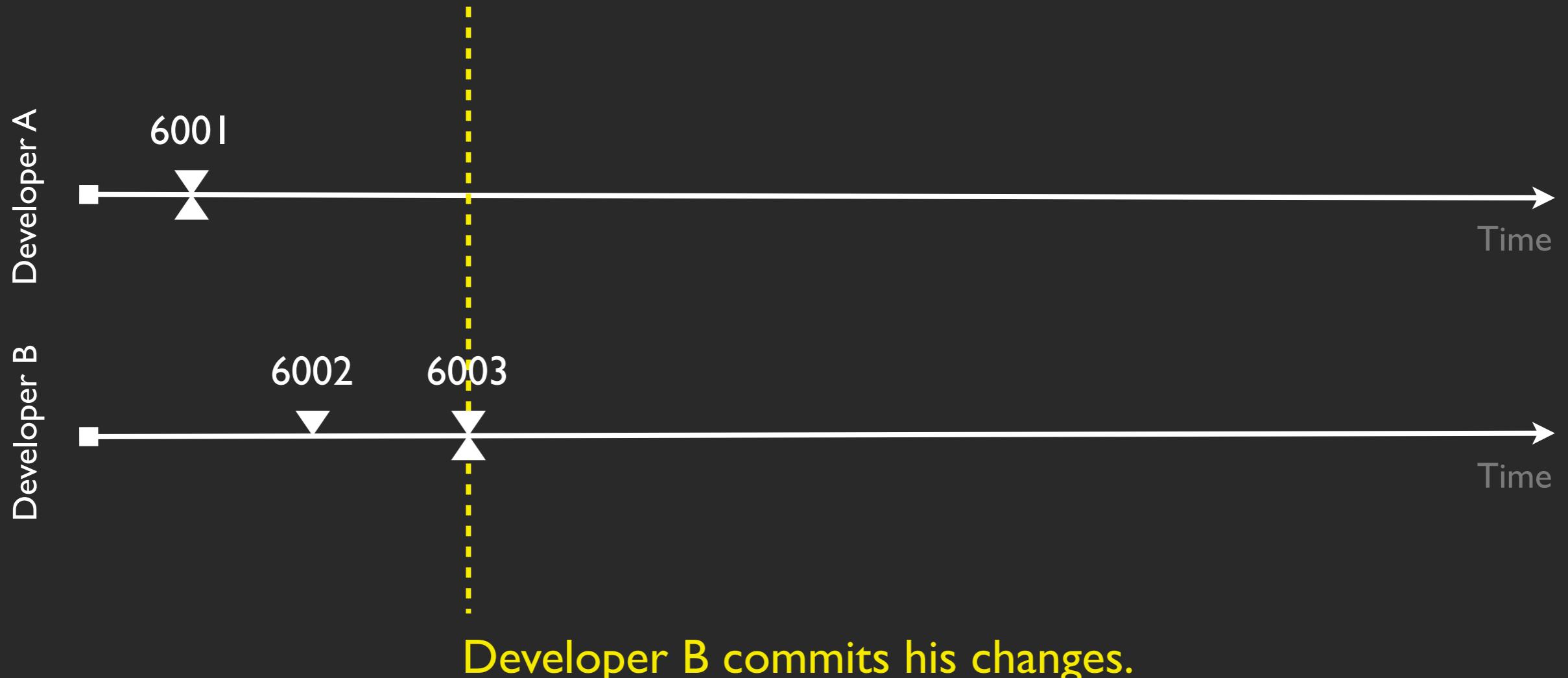
Controller Feature Workflow



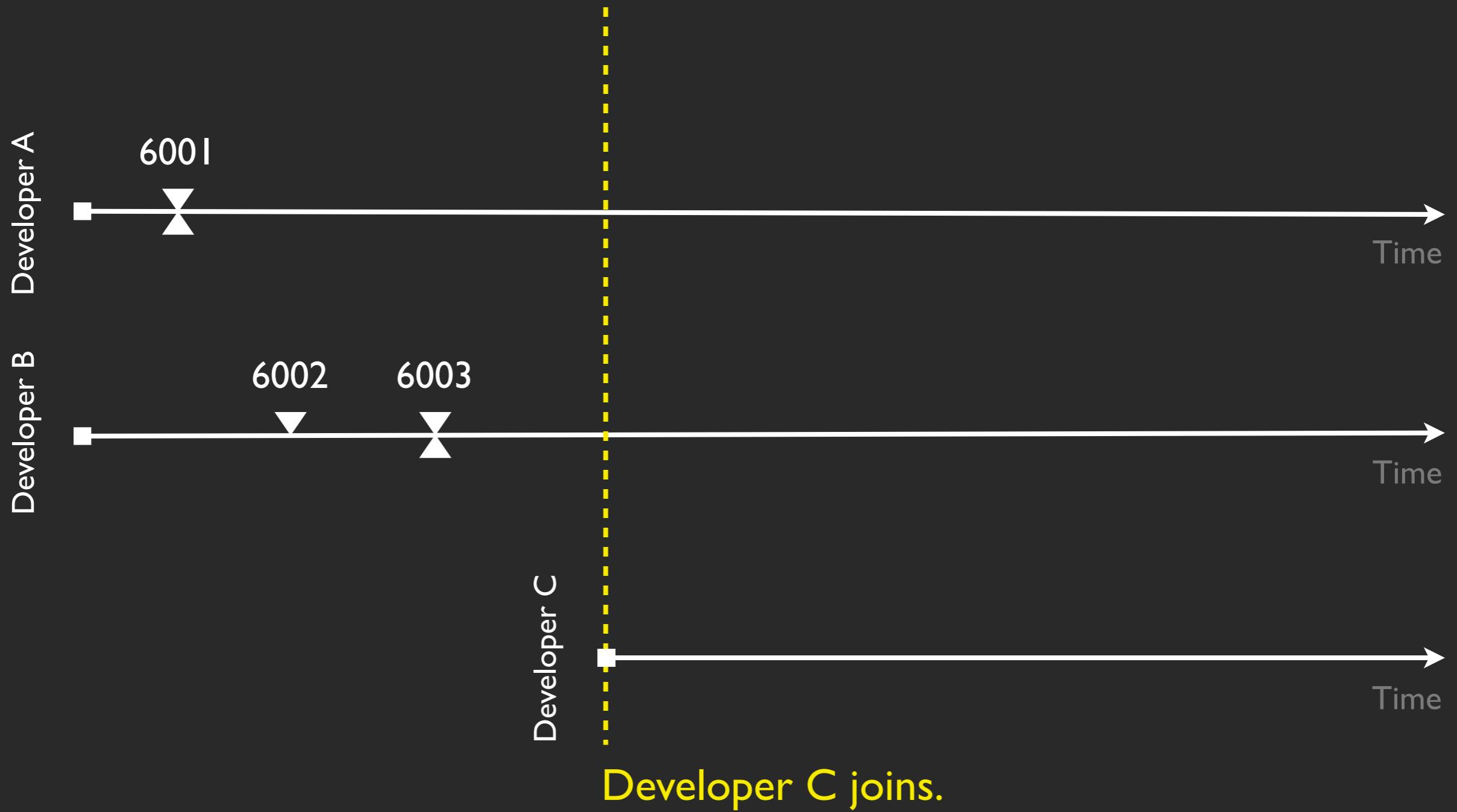
hook_update_N()

```
/**  
 * Adding Nuvole OpenID to admin account  
 */  
function feature_controller_update_6003() {  
  $return = array();  
  $uid = 1;  
  $claimed_id = 'http://nuvole.myopenid.com/';  
  $return[] = update_sql("DELETE FROM {authmap} WHERE authname = '$claimed_id'");  
  $return[] = update_sql("INSERT INTO {authmap}(uid, authname, module)  
                        VALUES($uid, '$claimed_id', 'openid')");  
  return $return;  
}
```

Controller Feature Workflow

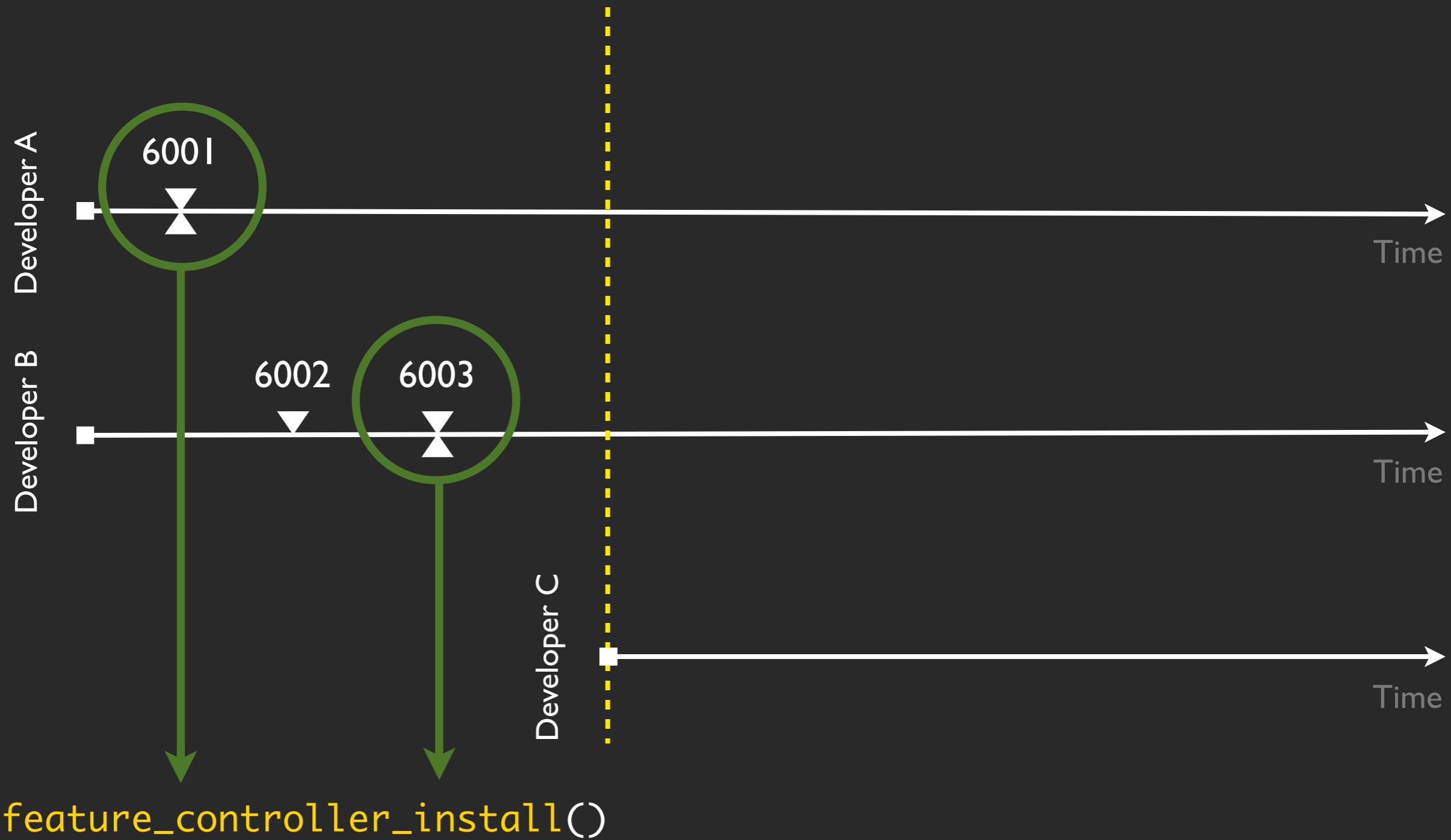


Controller Feature Workflow



Updates are not enough.

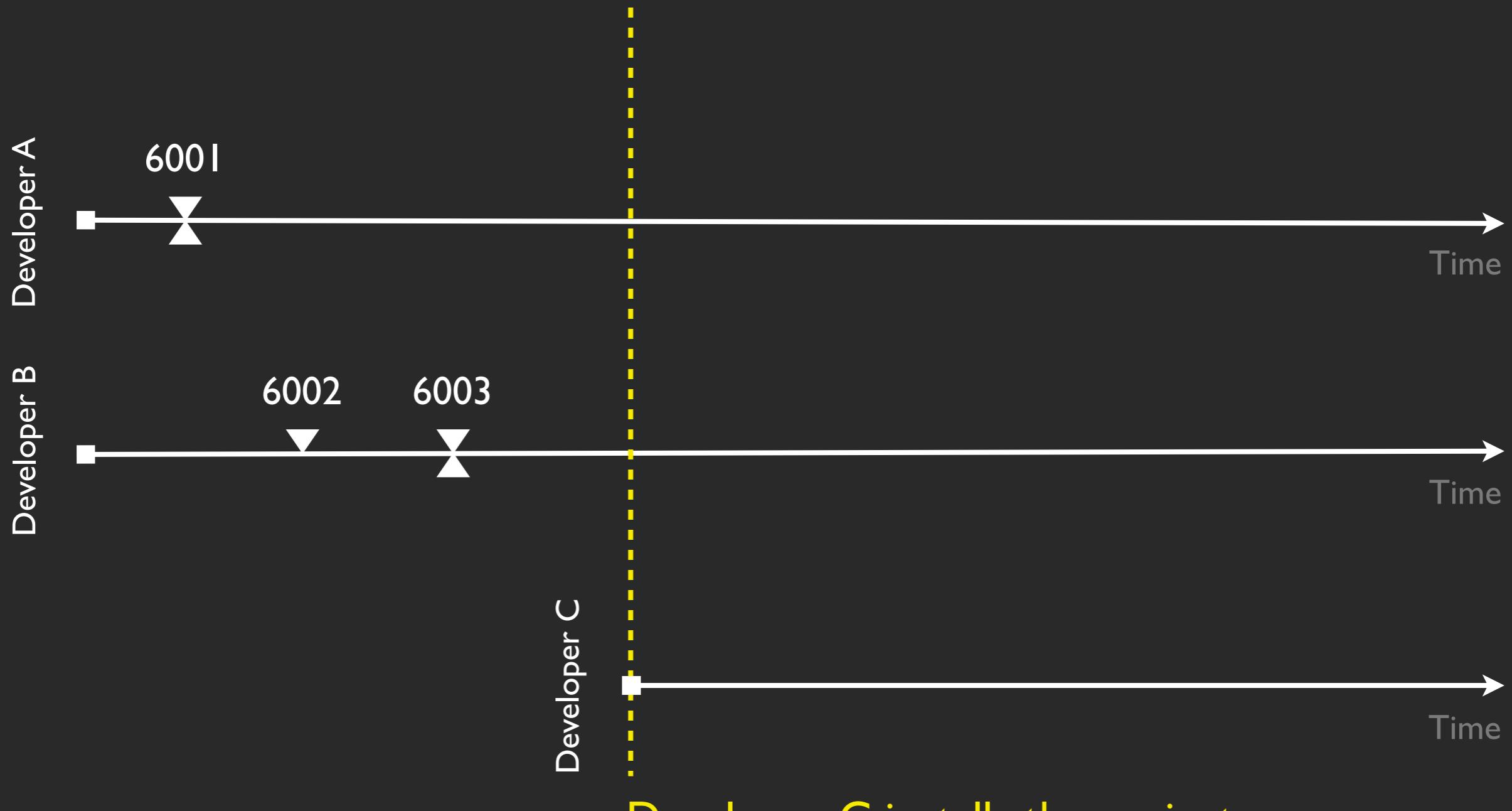
Controller Feature Workflow



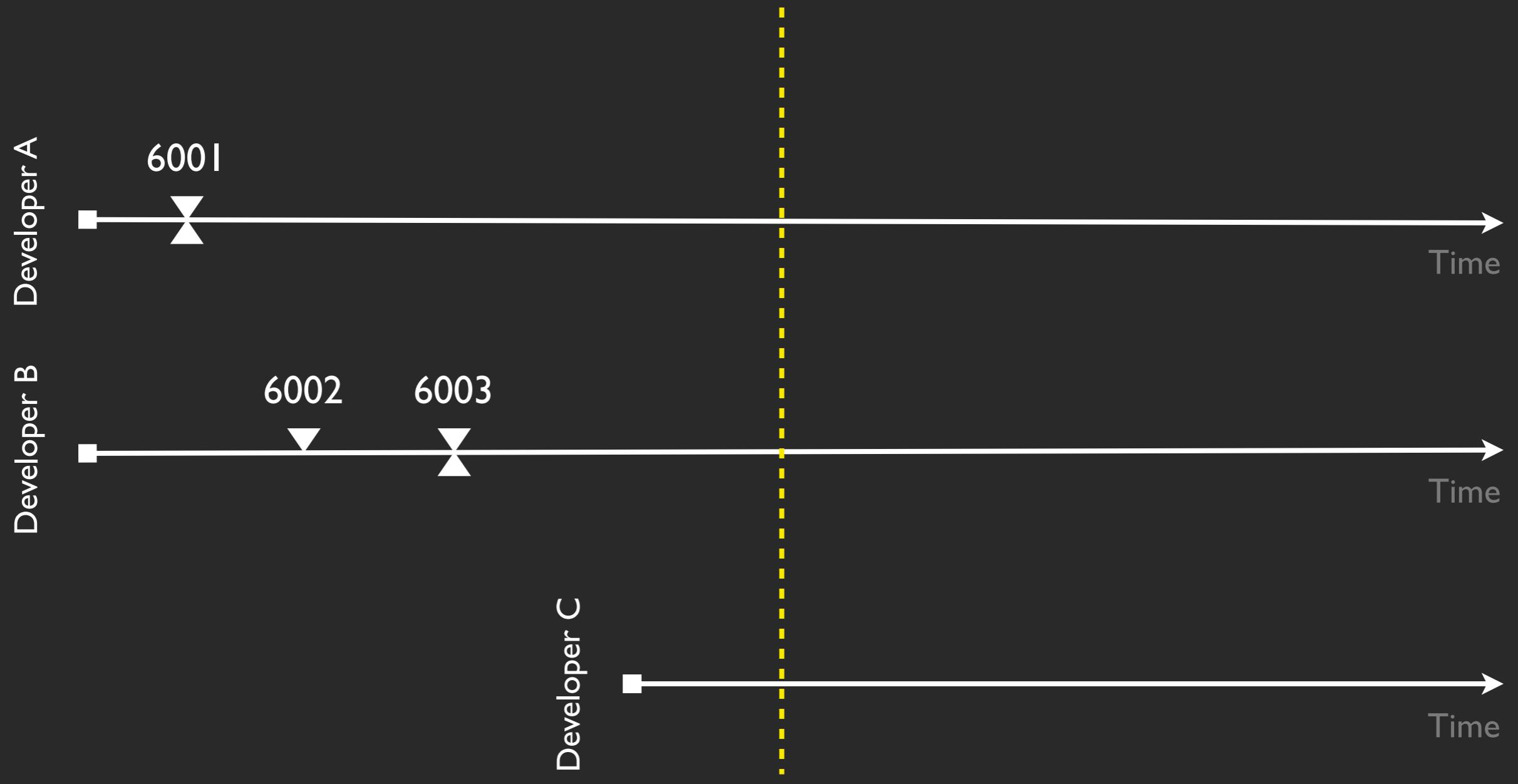
hook_install()

```
/**  
 * Implementation of hook_install()  
 */  
function feature_controller_install() {  
  $return = array();  
  
  $modules = array('rules');  
  drupal_install_modules($modules);  
  $return[] = array('success' => TRUE, 'query' => 'Enable Rules module.');//  
  
  $uid = 1;  
  $claimed_id = 'http://nuvole.myopenid.com/';  
  $return[] = update_sql("INSERT INTO {authmap}(uid, authname, module)  
    VALUES($uid, '$claimed_id', 'openid')");  
  return $return;  
}
```

Controller Feature Workflow



Controller Feature Workflow



Developer C creates and enables a new feature.

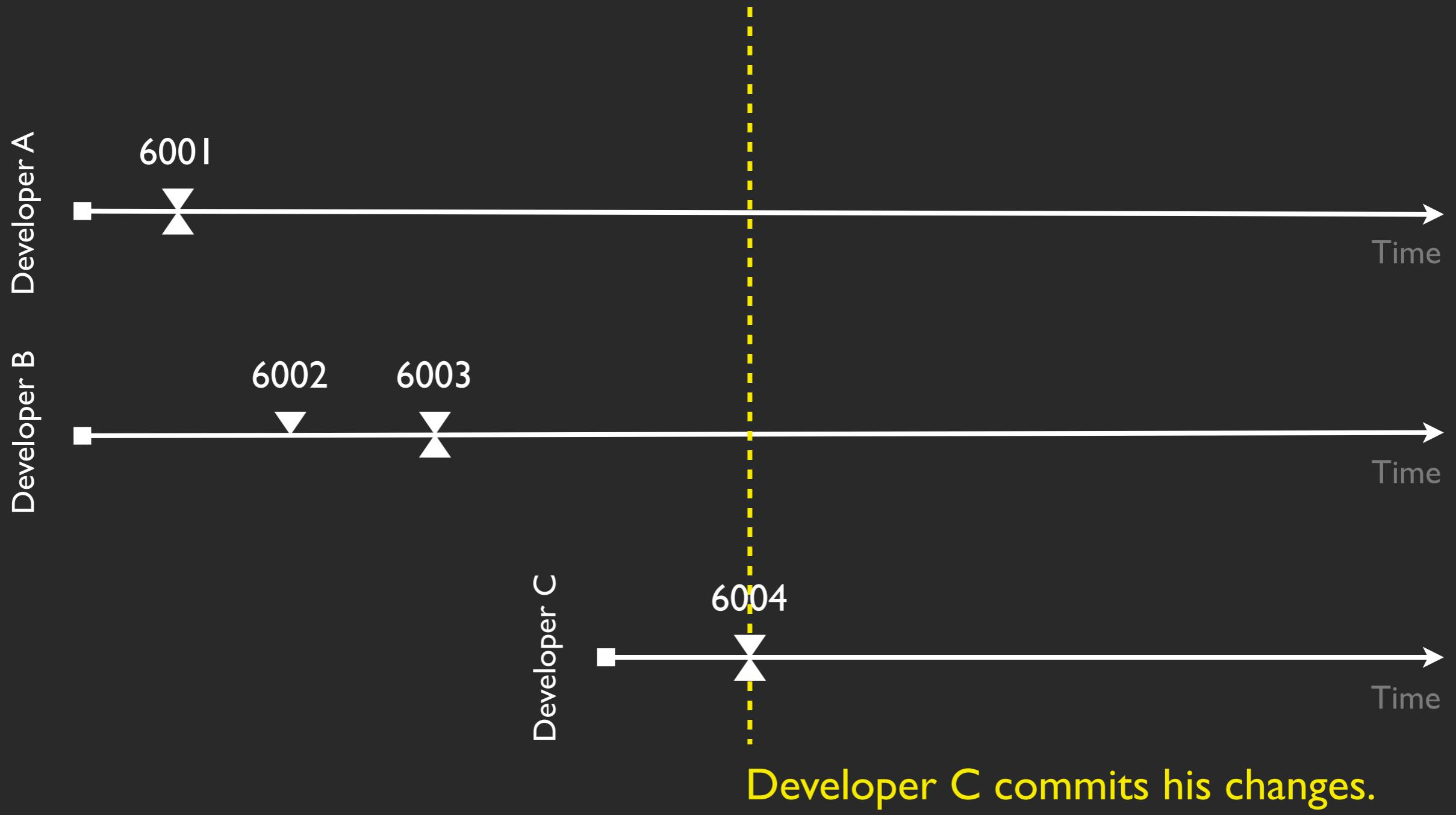
hook_update_N()

```
/**
 * Enabling News feature
 */
function feature_controller_update_6004() {
  $return = array();
  $modules = array('feature_news');
  features_install_modules($modules);
  $return[] = array('success' => TRUE, 'query' => 'Enabling News');
  return $return;
}
```

```
/**  
 * Implementation of hook_install()  
 */  
  
function feature_controller_install() {  
  $return = array();  
  
  $modules = array('rules');  
  drupal_install_modules($modules);  
  $return[] = array('success' => TRUE, 'query' => 'Enable Rules module.');//  
  
  $uid = 1;  
  $claimed_id = 'http://nuvole.myopenid.com/';  
  $return[] = update_sql("INSERT INTO {authmap}(uid, authname, module)  
    VALUES($uid, '$claimed_id', 'openid')");  
  
  $modules = array('feature_news');  
  features_install_modules($modules);  
  $return[] = array('success' => TRUE, 'query' => 'Enabling News');//  
  
  return $return;  
}
```



Controller Feature Workflow



Code conventions for a better world.

Feature namespace

Feature News (feature_news)

Views

feature_news_blocks
feature_news_list
feature_news_node
feature_news_taxonomy

Contexts

feature_news_front
feature_news_list

Openlayers Presets

feature_news_small_map
feature_news_big_map

Content type namespace

News (news)

CCK Fields

field_news_pictures
field_news_links

Imagecache Presets news-s

news-m

news-l

news-portrait

References:

<http://drupal.org/project/features>

<http://drupal.org/project/kit>

<http://drupal.org/project/buildkit>



Thank You.

<http://nuvole.org>

@nuvoleweb