

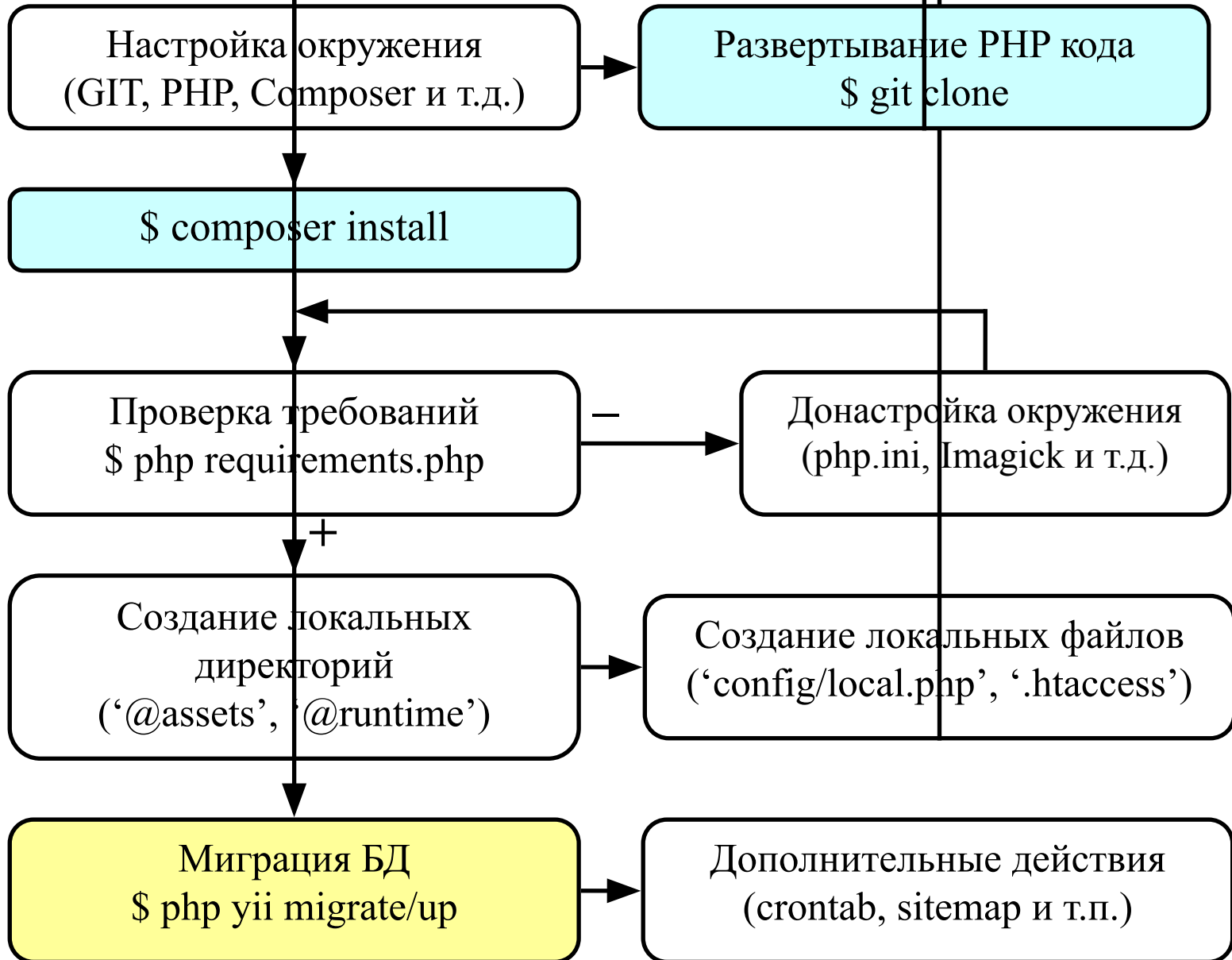
# Миграции и Инсталляции



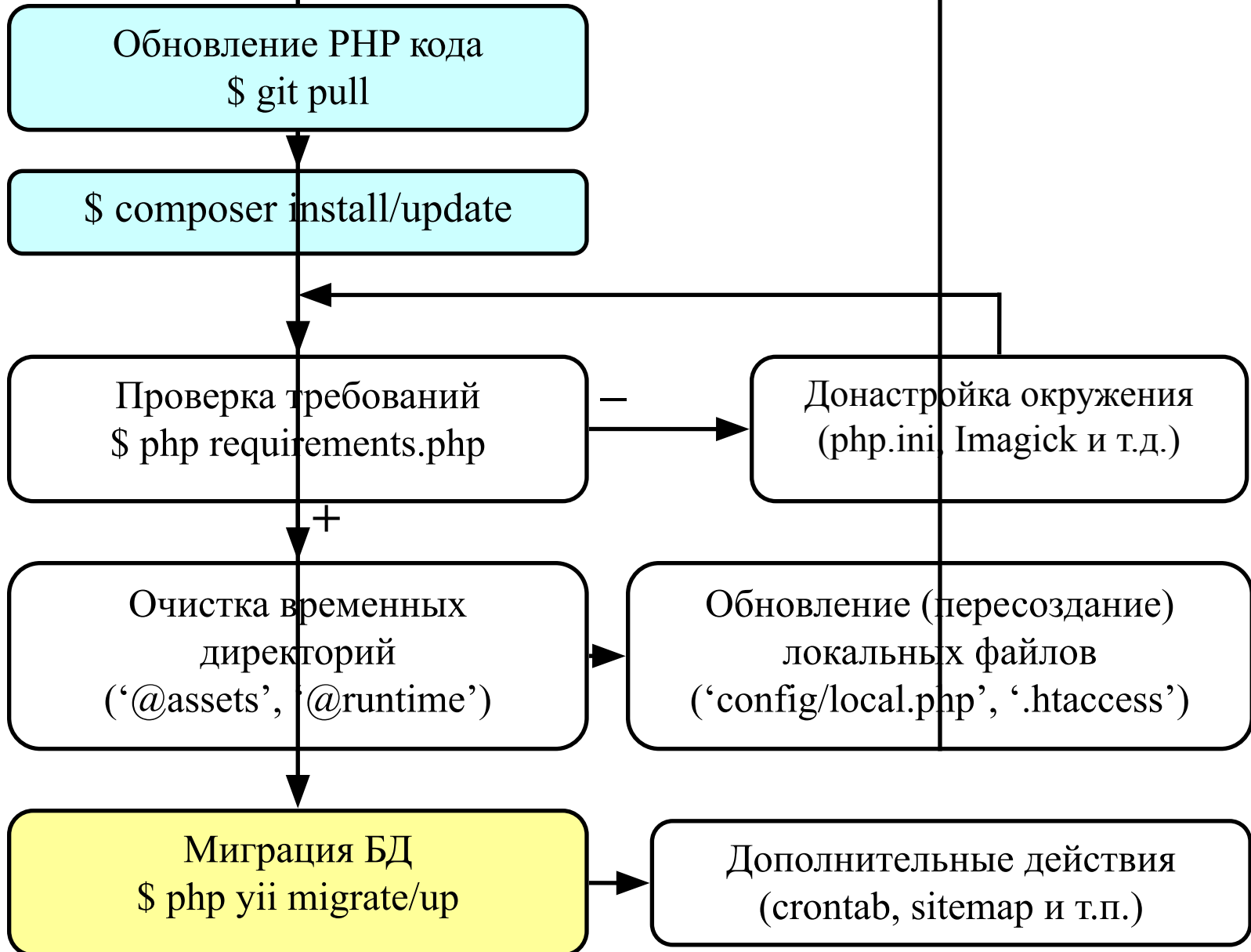
Климов П.В.

QuartSoft  
YiiSoft

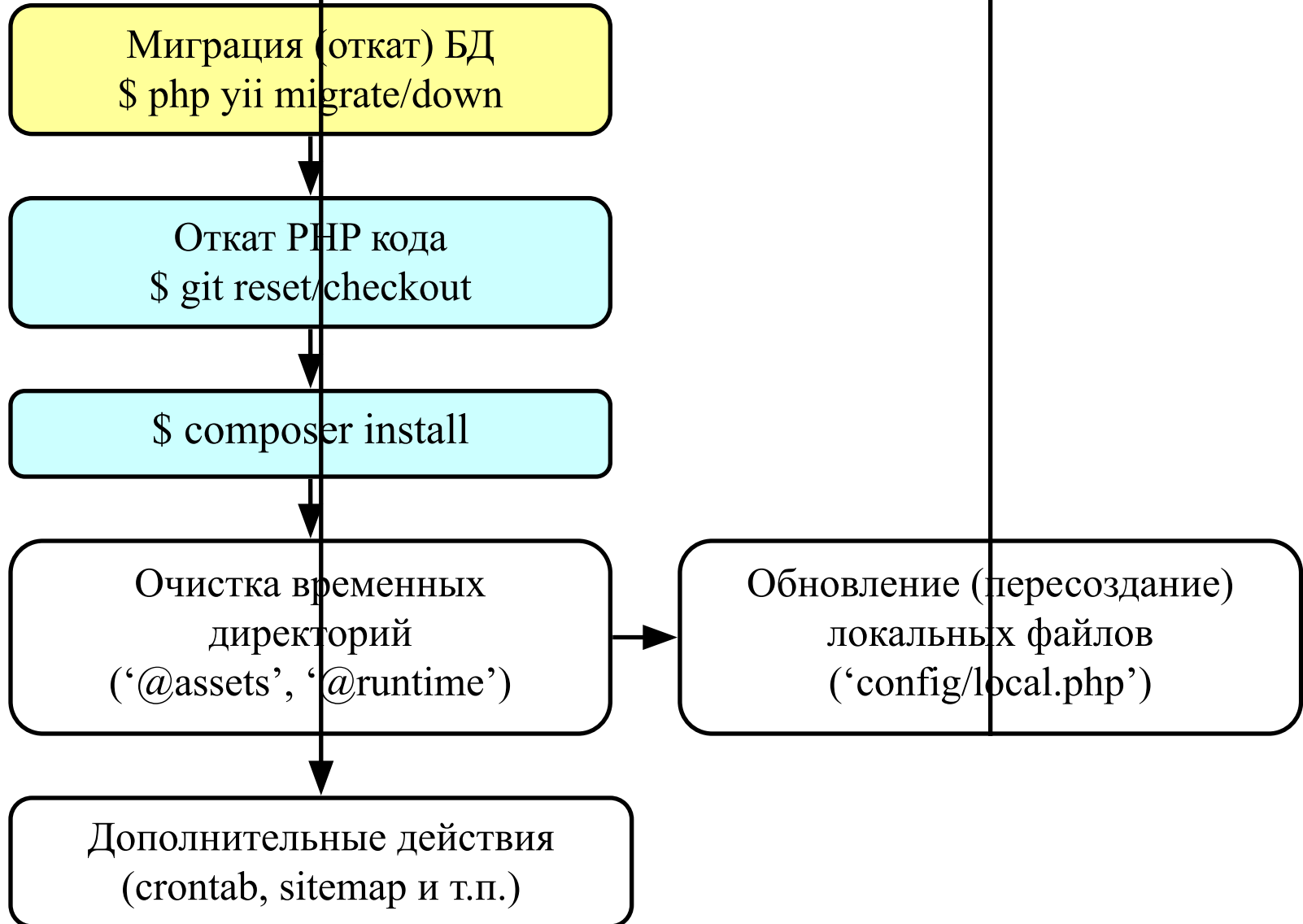
# Инсталляция рабочей копии проекта



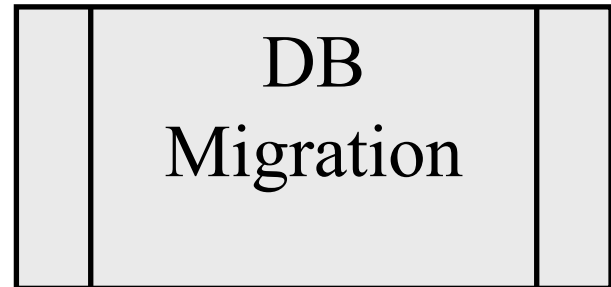
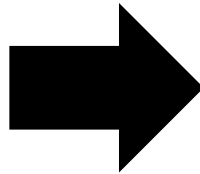
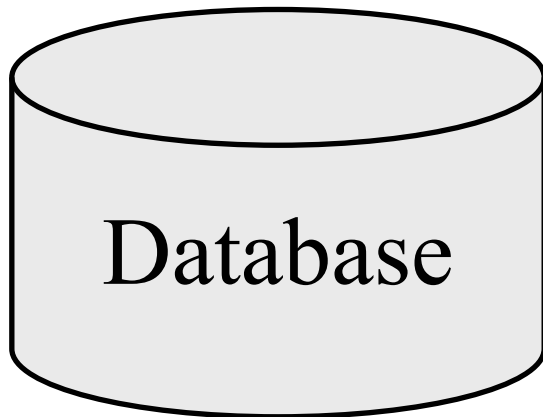
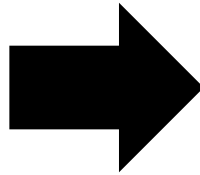
# Обновление рабочей копии проекта



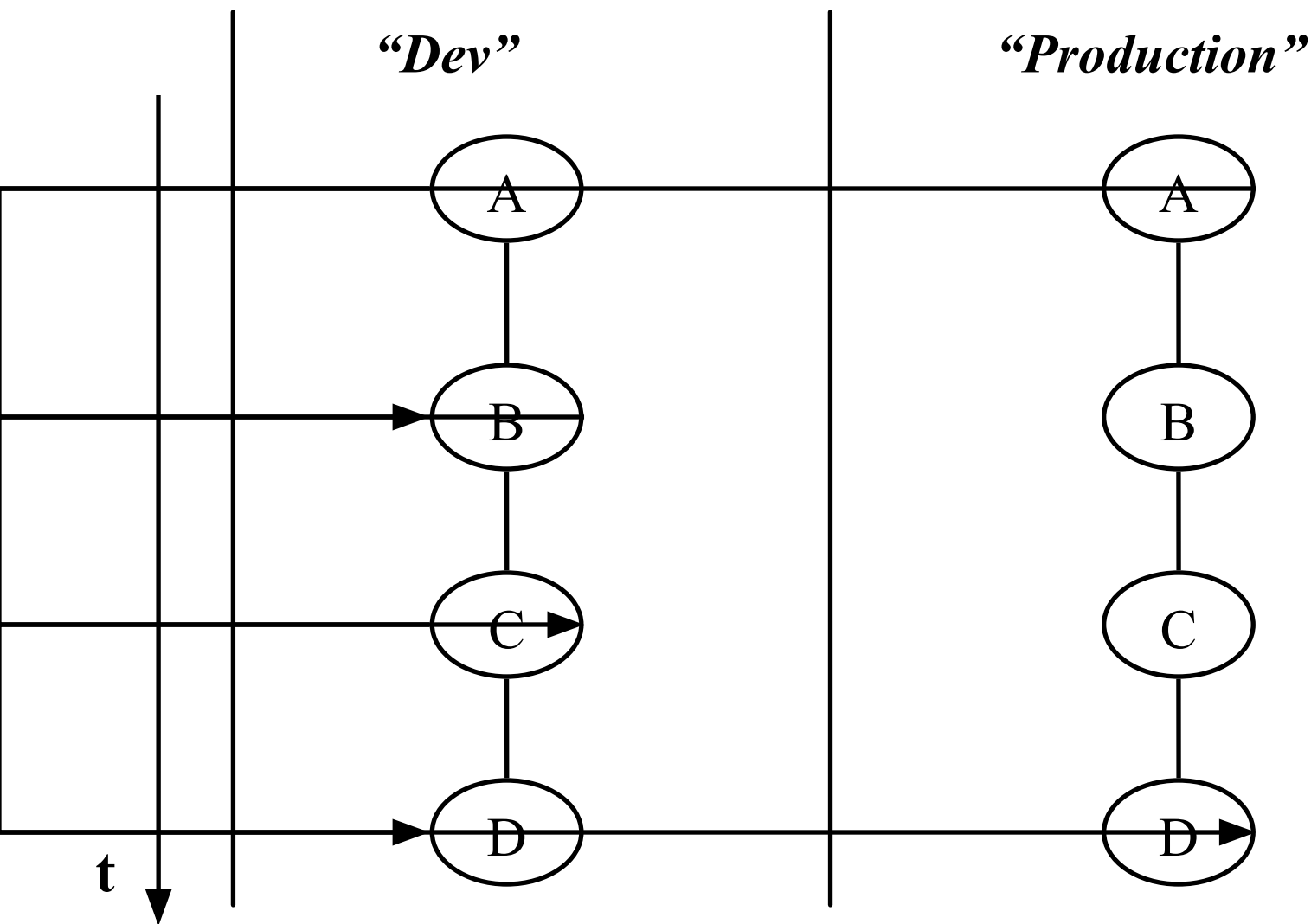
# Откат рабочей копии проекта



# Системы контроля версий



# Миграции по разным рабочим копиям





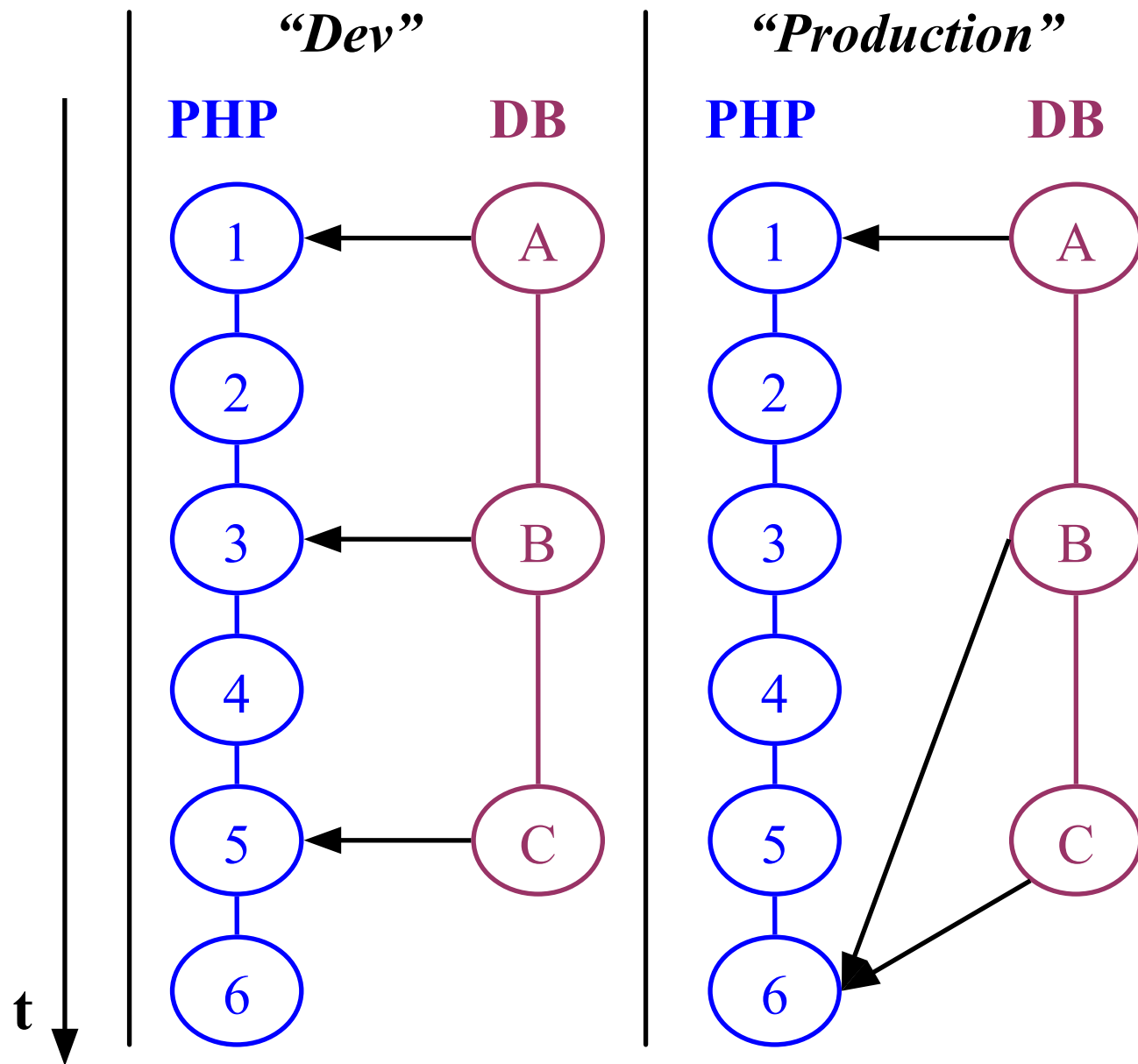
```
class m170506_185632_createUser extends Migration {  
    public function up() {  
        $this->createTable('User', []);  
    }  
    public function down() {  
        echo "m???_createUser cannot be reverted.\n";  
        return false;  
    }  
}
```

---



```
class m170506_185632_createUser extends Migration {  
    public function up() {  
        $this->createTable('User', []);  
    }  
    public function down() {  
        $this->dropTable('User');  
    }  
}
```

# История РНР и история БД







```
class m141106_185632_log_init extends Migration {  
    public function up() {  
        foreach (Yii::$app->getLog()->targets as $target) {  
            if ($target instanceof DbTarget) {  
                $this->createTable($target->logTable, [...]);  
            }  
        }  
    }  
}
```



---

```
class m141106_185632_log_init extends Migration {  
    public function up() {  
        $columns = [...];  
        $this->createTable('AppLog', $columns);  
        $this->createTable('ErrorLog', $columns);  
    }  
}
```



```
use app\models\User;
```

```
class m141106_185632_updateUser extends Migration {  
    public function up() {  
        User::updateAll(['statusId' => 5], ['statusId' => 10]);  
    }  
    public function down() {  
        foreach (User::find()->where(['statusId' => 10])->all() as $user) {  
            $user->statusId = 5;  
        }  
    }  
}
```



---

```
class m141106_185632_updateUser extends Migration {  
    public function up() {  
        $this->update('User', ['statusId' => 5], ['statusId' => 10]);  
    }  
}
```



```
class m141106_185632_createFaqCategory extends Migration {  
    public function up() {  
        $this->createTable('FaqCategory', [...]);  
    }  
}
```

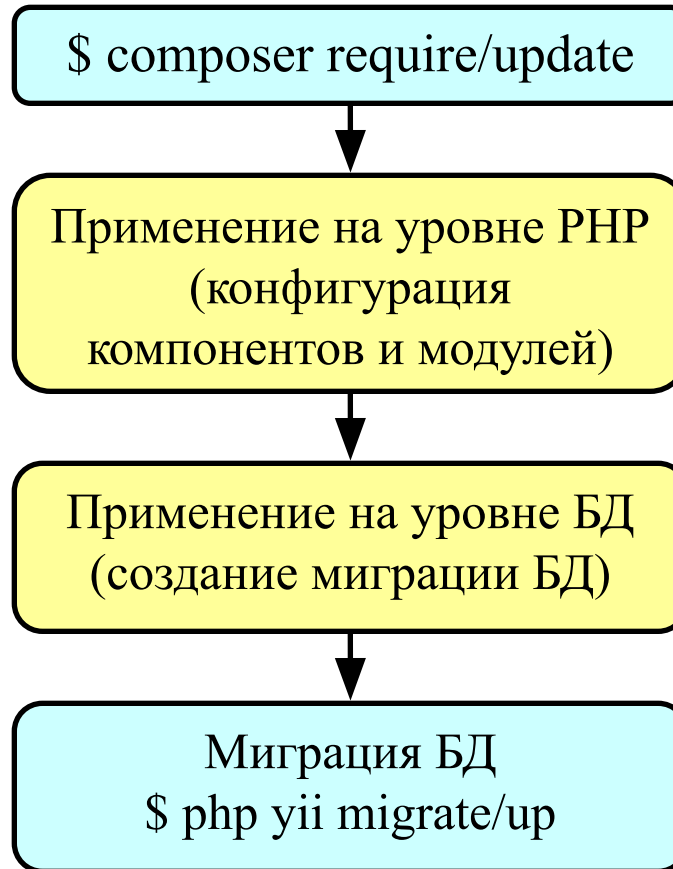
```
class m141106_185632_createFaqQuestion extends Migration {  
    public function up() {  
        $this->createTable('FaqQuestion', [...]);  
    }  
}
```



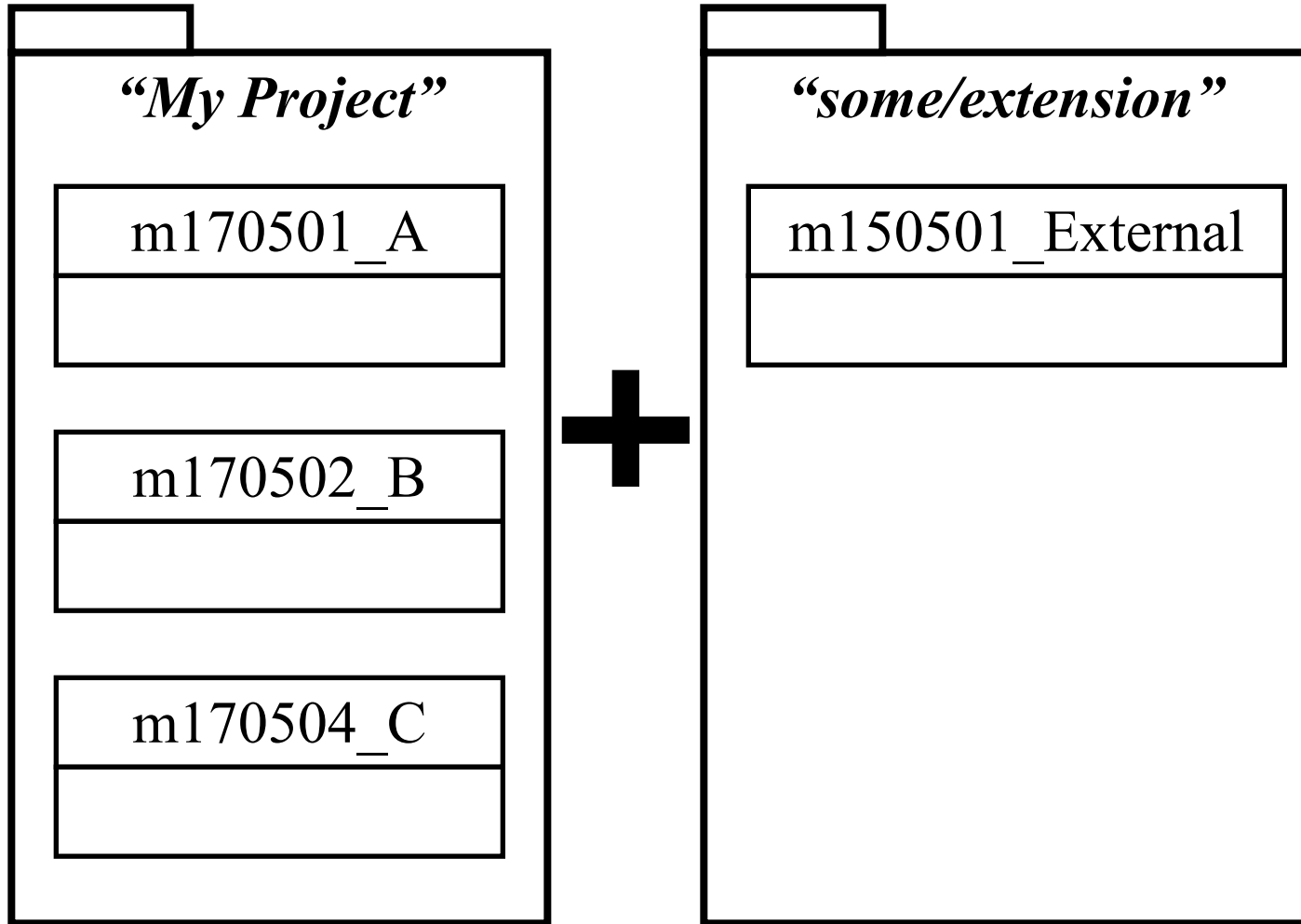
---

```
class m141106_185632_createFaq extends Migration {  
    public function up() {  
        $this->createTable('FaqCategory', [...]);  
        $this->createTable('FaqQuestion', [...]);  
    }  
}
```

# «Инсталляция» расширения / модуля



# Применение миграций из нескольких ИСТОЧНИКОВ



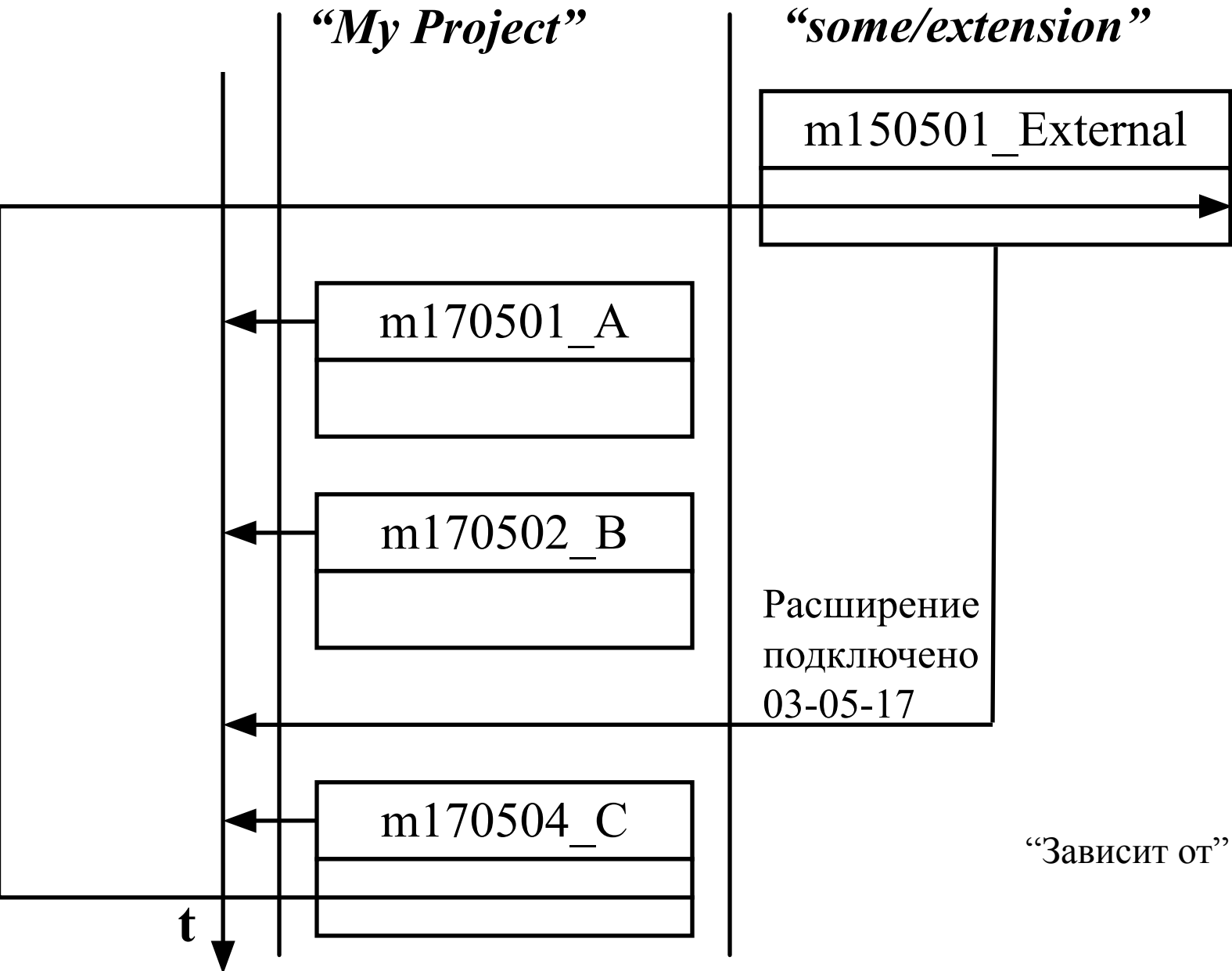
# Переключение «migrationPath»

```
$ php yii migrate
--migrationPath=@app/migrations
$ php yii migrate
--migrationPath=@some/extension/mi
```

## Использование пространства имен

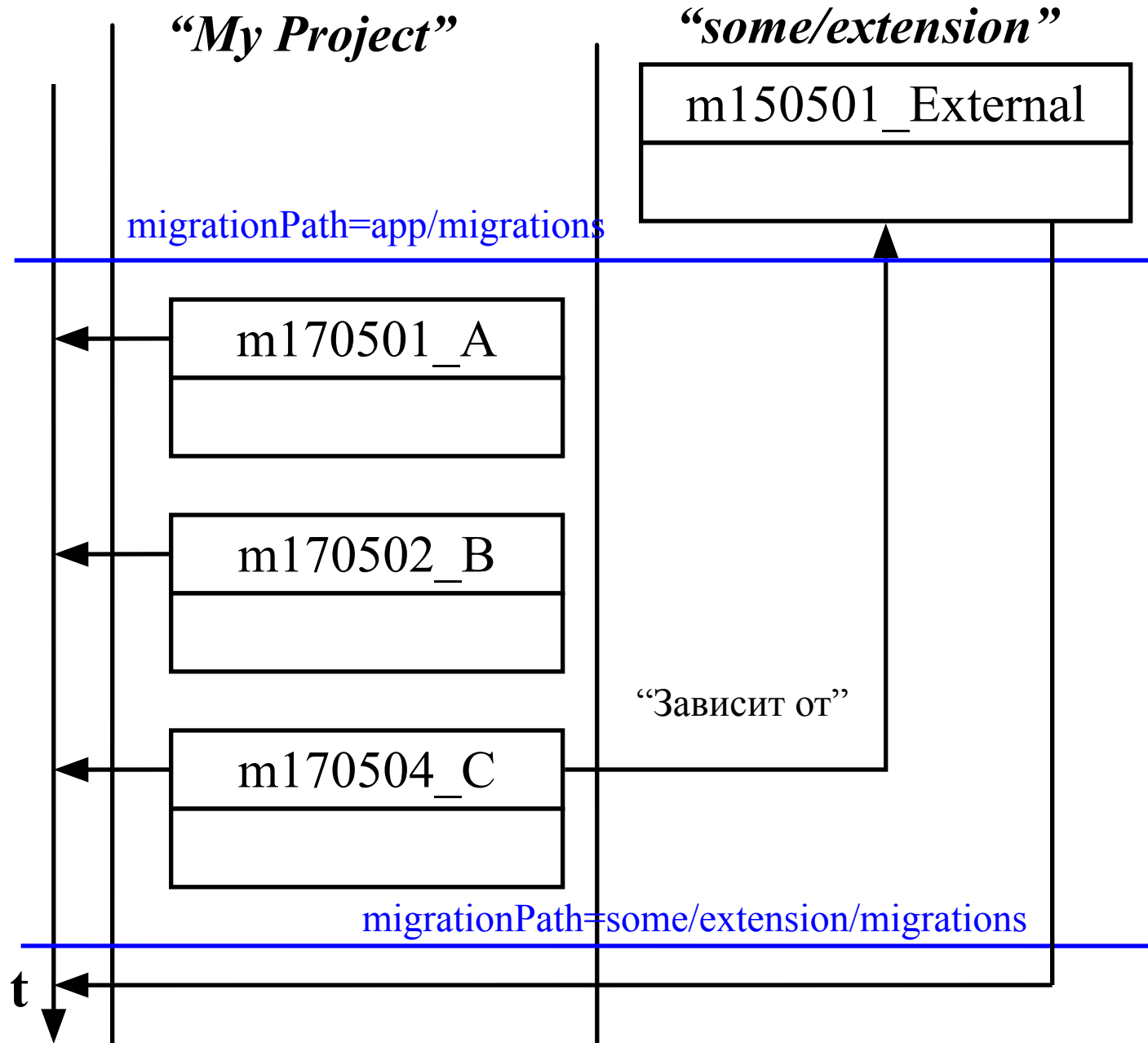
```
return [
    'controllerMap' => [
        'migrate' => [
            'class' => 'yii\console\controllers\MigrateController',
            'migrationNamespaces' => [
                'app\migrations',
                'some\extension\migrations',
            ],
        ],
    ],
    // ...
];
```

# История на сервере разработки



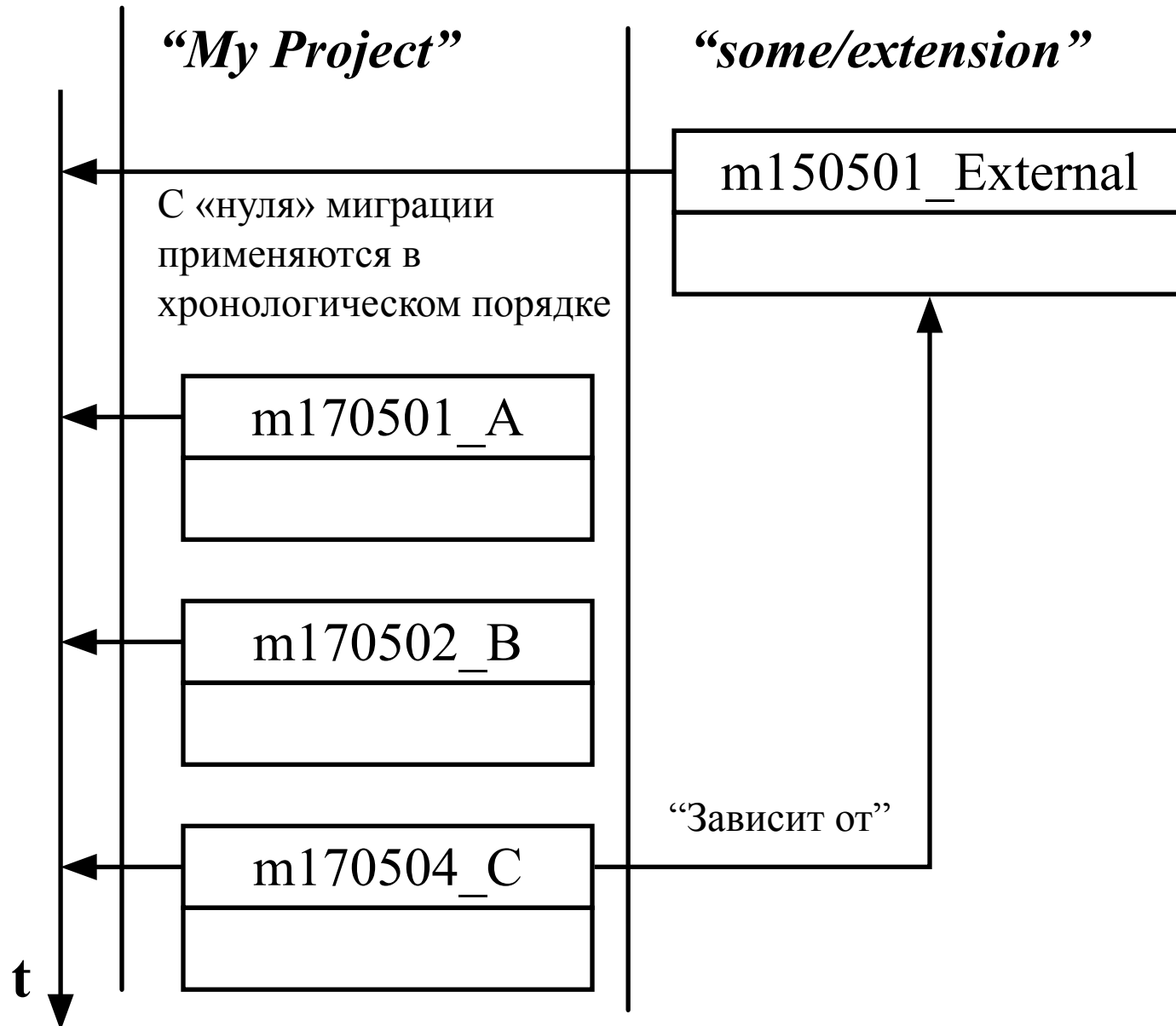
“Зависит от”

# История на «production» сервере





# История на «production» сервере



# Повторное использование кода

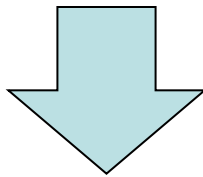
```
$array = [$day1Revenue, $day2Revenue, $day3Revenue];
```

```
$avgRevenue = array_sum($array) / count($array);
```

```
// ...
```

```
$array = [$day1Costs, $day2Costs, $day3Costs];
```

```
$avgCosts = array_sum($array) / count($array);
```



```
function avg (array $values) {  
    return array_sum($values) / count($values);  
}
```

```
$avgRevenue = avg([$day1Revenue, $day2Revenue, $day3Revenue]);
```

```
// ...
```

```
$avgCosts = avg([$day1Costs, $day2Costs, $day3Costs]);
```

# Рефакторинг

```
function avg (array $values) {  
    if (empty($values)) {  
        return 0;  
    }  
    return array_sum($values) / count($values);  
}
```

```
$avgRevenue = avg([$day1Revenue, $day2Revenue, $day3Revenue]);
```

```
// ...
```

```
$avgCosts = avg([$day1Costs, $day2Costs, $day3Costs]);
```

# Внешняя миграция, ревизия №1

```
class m160201_132117_someExtensionMigration extends Migration
{
    public function up()
    {
        $this->createTable('BlogPostCategory', [...]);
        $this->createTable('BlogPost', [...]);
    }

    public function down()
    {
        $this->dropTable('BlogPost');
        $this->dropTable('BlogPostCategory');
    }
}
```

# Внешняя миграция, ревизия №2

```
class m160201_132117_someExtensionMigration extends Migration
{
    public function up()
    {
        $this->createTable('BlogPostCategory', [...]);
        $this->createTable('BlogPost', [...]);
        $this->createTable('BlogPostComment', [...]);
    }

    public function down()
    {
        $this->dropTable('BlogPostComment');
        $this->dropTable('BlogPost');
        $this->dropTable('BlogPostCategory');
    }
}
```

# Раздельная история

```
$ php yii migrate  
--migrationPath=@app/migrations  
$ php yii migrate  
--migrationPath=@some/forum/migrations
```

```
return [  
    'controllerMap' => [  
        'migrate-forum' => [  
            'class' => 'yii\console\controllers\MigrateController',  
            'migrationNamespaces' => [  
                'some\forum\migrations',  
            ],  
            'migrationTable' => 'migration_module',  
        ],  
    ],  
    // ...  
];
```

# Наследование

*// Без пространства имен:*

```
require (Yii::getAlias('@some/extension/  
m160201_132117_someExtensionMigration.php'));
```

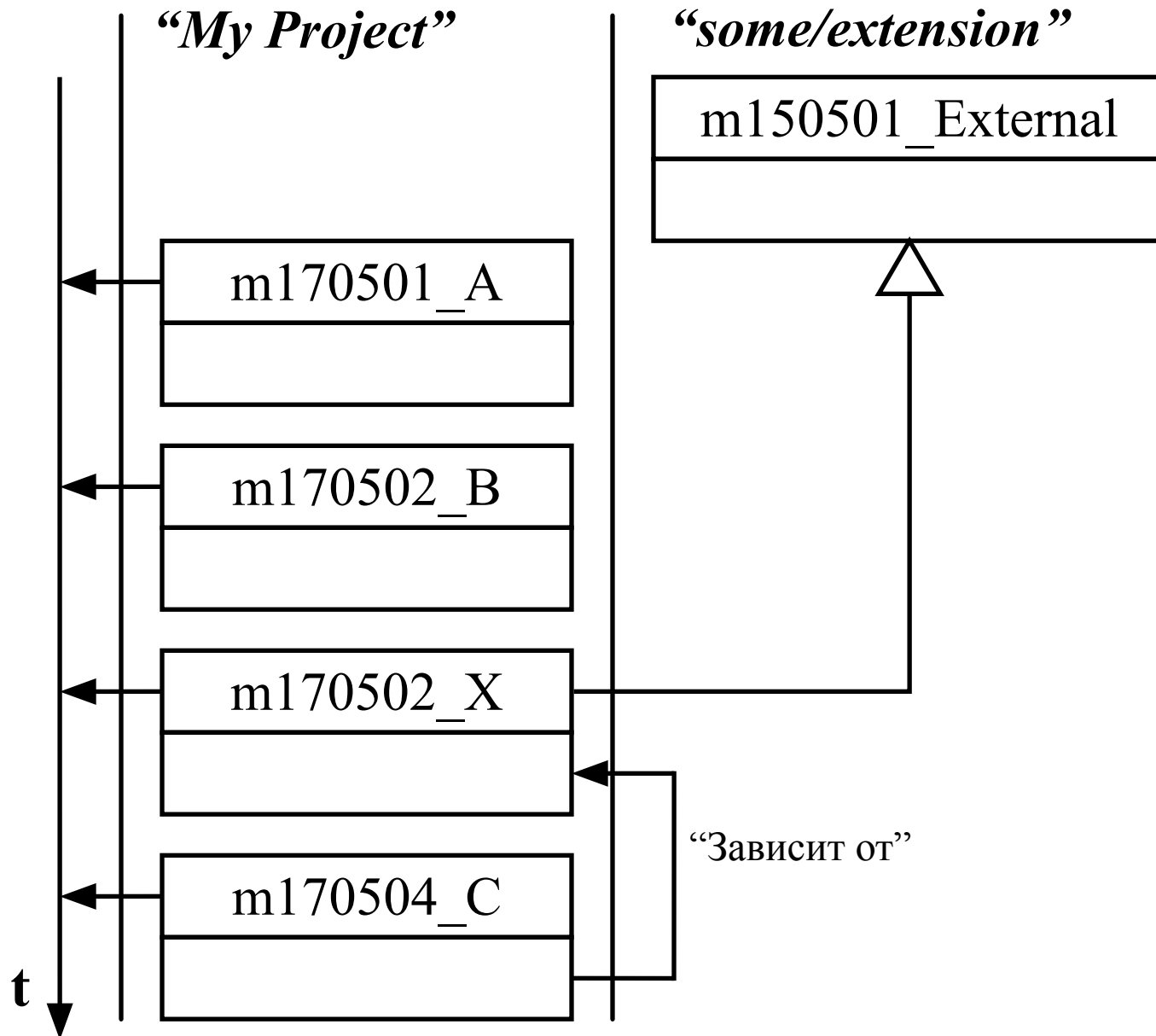
```
class m170505_142134_applySomeExtension extends  
m160201_132117_someExtensionMigration {}
```

*// С пространством имен:*

```
namespace app\migrations;
```

```
class M170505142134ApplySomeExtension extends  
\some\extension\migrations\M160201132117SomeExtensionMigration {}
```

# История при наследовании





# Наследование и модификация

```
require (Yii::getAlias('@yii/rbac/m140506_102106_rbac_init.php'));
```

```
class m170505_142134_rbac extends m140506_102106_rbac_init
```

```
{
```

```
    protected function getAuthManager()
```

```
    {
```

```
        return new yii\rbac\DbManager(['itemTable' => 'RbacItem', ...]);
```

```
    }
```

```
    public function up()
```

```
    {
```

```
        parent::up();
```

```
        $this->alterColumn($this->getAuthManager()->assignmentTable,
```

```
            'user_id', $this->integer()->notNull());
```

```
    }
```

```
}
```

# Инверсия

```
require (Yii::getAlias('@yii/rbac/m140506_102106_rbac_init.php'));
```

```
class m170505_142134_undoRbac extends m140506_102106_rbac_init
{
    protected function getAuthManager()
    {
        return new yii\rbac\DbManager(['itemTable' => 'RbacItem', ...]);
    }
    public function up()
    {
        parent::down(); // down -> up
    }
    public function down()
    {
        parent::up(); // up -> down
    }
}
```

# Агрегация

```
require (Yii::getAlias('@some/extension/m160201_132117_A.php'));  
require (Yii::getAlias('@another/extension/m150301_141133_B.php'));
```

```
class m170505_142134_aggregation extends \yii\db\Migration  
{  
    public function up()  
    {  
        (new m160201_132117_A(['db' => $this->db]))->up();  
        (new m150301_141133_B(['db' => $this->db]))->up();  
    }  
    public function down()  
    {  
        (new m150301_141133_B(['db' => $this->db]))->down();  
        (new m160201_132117_A(['db' => $this->db]))->down();  
    }  
}
```

# Агрегация с пространством имен

```
namespace app\migrations;
```

```
use some\extension\M160201132117A.php';
```

```
use another\extension\M150301141133B.php;
```

```
class M170505142134Aggregation extends \yii\db\Migration
```

```
{
```

```
    public function up()
```

```
    {
```

```
        (new M160201132117A(['db' => $this->db]))->up();
```

```
        (new M150301141133B(['db' => $this->db]))->up();
```

```
    }
```

```
    public function down()
```

```
    {
```

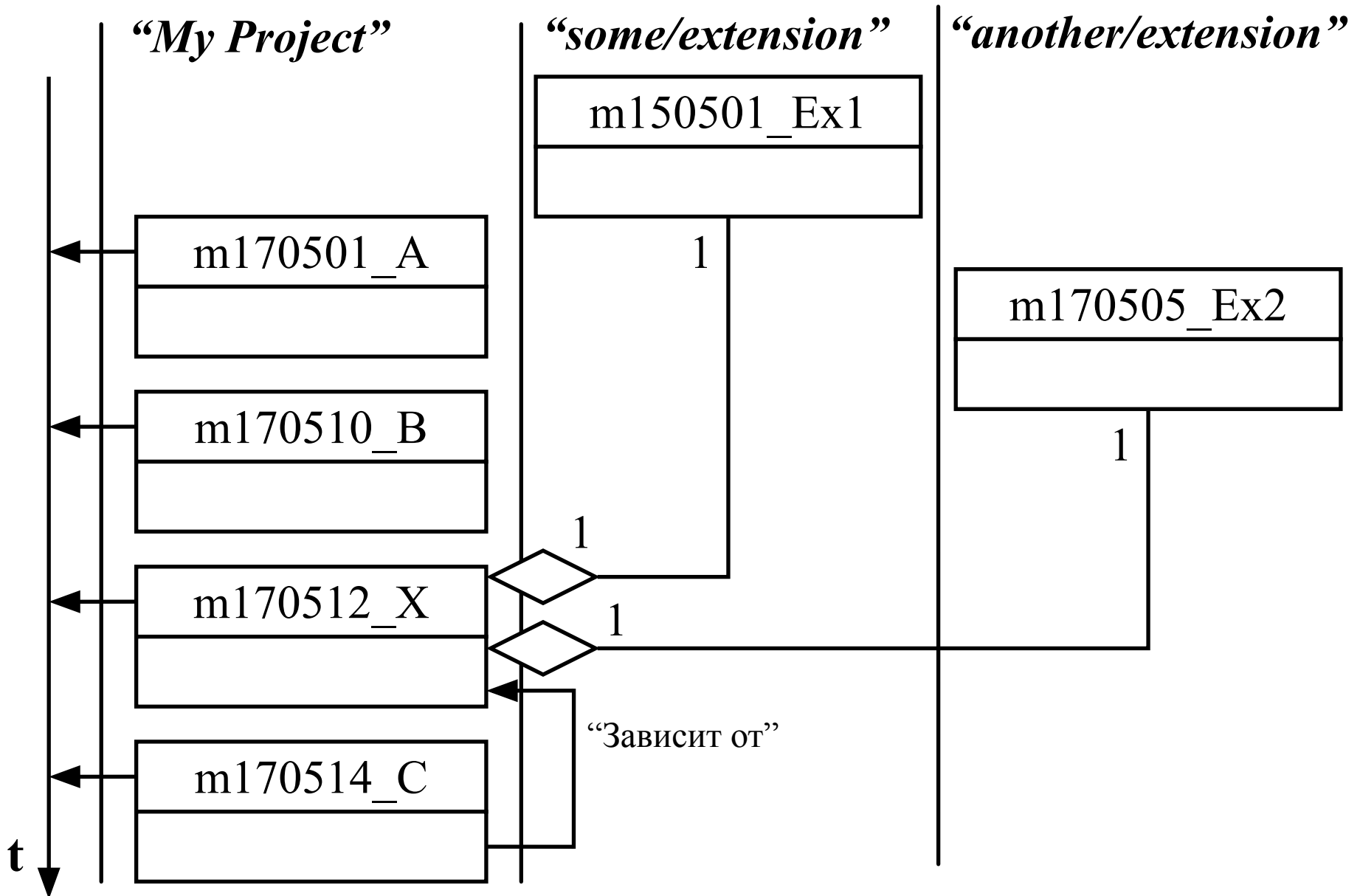
```
        (new M150301141133B(['db' => $this->db]))->down();
```

```
        (new M160201132117A(['db' => $this->db]))->down();
```

```
    }
```

```
}
```

# История при агрегации



# Независимое использование миграций

```
use my\extension\M170505142134Foo;
```

```
class InstallController extends \yii\console\Controller
```

```
{
```

```
    public function actionInstall()
```

```
    {
```

```
        (new M170505142134Foo(['db' => Yii::$app->db]))->up();
```

```
    }
```

```
    public function actionUninstall()
```

```
    {
```

```
        (new M170505142134Foo(['db' => Yii::$app->db]))->down();
```

```
    }
```

```
}
```

# Миграции в модульных тестах

```
class BlogPostTest extends TestCase
{
    protected function setUp()
    {
        $this->mockApplication();
        Yii::$app->db->beginTransaction();
        (new M170505142134Foo(['db' => Yii::$app->db]))->safeUp();
    }

    protected function tearDown()
    {
        Yii::$app->db->getTransaction()->rollback();
    }

    // ...
}
```

# Миграции и Инсталляции

- Инсталляция, обновление, откат рабочей копии проекта
- Миграции – VCS для БД
- Типичные ошибки
- Применение миграций из нескольких источников
- Независимое использование миграций