Yii2. Upload
Git

sd@freematiq.com
https://github.com/freematiq/php

Форма

http://www.yiiframework.com/doc-2.0/guide-tutorial-core-validators.html

SubscribeForm.php

```php
/* @var UploadedFile */
public $file = null;


['file', 'file'],


<?php $form = ActiveForm::begin([
        'action' => ['/subscribe'],
        'options' => [
            'method' => 'post',
            'enctype' => 'multipart/form-data',
        ]
    ]
); ?>
```

Validation

Для проверки нужно создать объект

```
$form->file = UploadedFile::getInstance($form, ‘file');
```

Дополнительные проверки в форме

```
[['file'], 'file', 'skipOnEmpty' => false],
[['file'], 'file', 'extensions' => 'gif, jpg’],
```

Параметры php.ini

upload_max_filesize - ограничение на размер файла
post_max_size - ограничение на размер POST запроса

PHPDoc

Описание методов и параметров

```php
/**
 * Finds the User model based on its primary key value.
 * If the model is not found, a 404 HTTP exception will be thrown.
 * @param integer $id
 * @return User the loaded model
 * @throws NotFoundHttpException if the model cannot be found
 */
protected function findModel($id)
```

Описание переменных для подсказок

```php
/* @var $this yii\web\View */
/* @var $model app\models\Course */
```

Наследование

```php
/**
 * @inheritdoc
 */
```

Несколько типов

```php
* @param string|array $url the URL to be redirected to. This can be in one of
the following formats:
$this->redirect();
```

ActiveQuery

Model::find() // вернёт ActiveQuery
->where() // добавить условие
->andWhere(...) - // составное условие
->orWhere(...) - // составное условие
->groupby(...) - //
->count() // вернуть число записей

Git

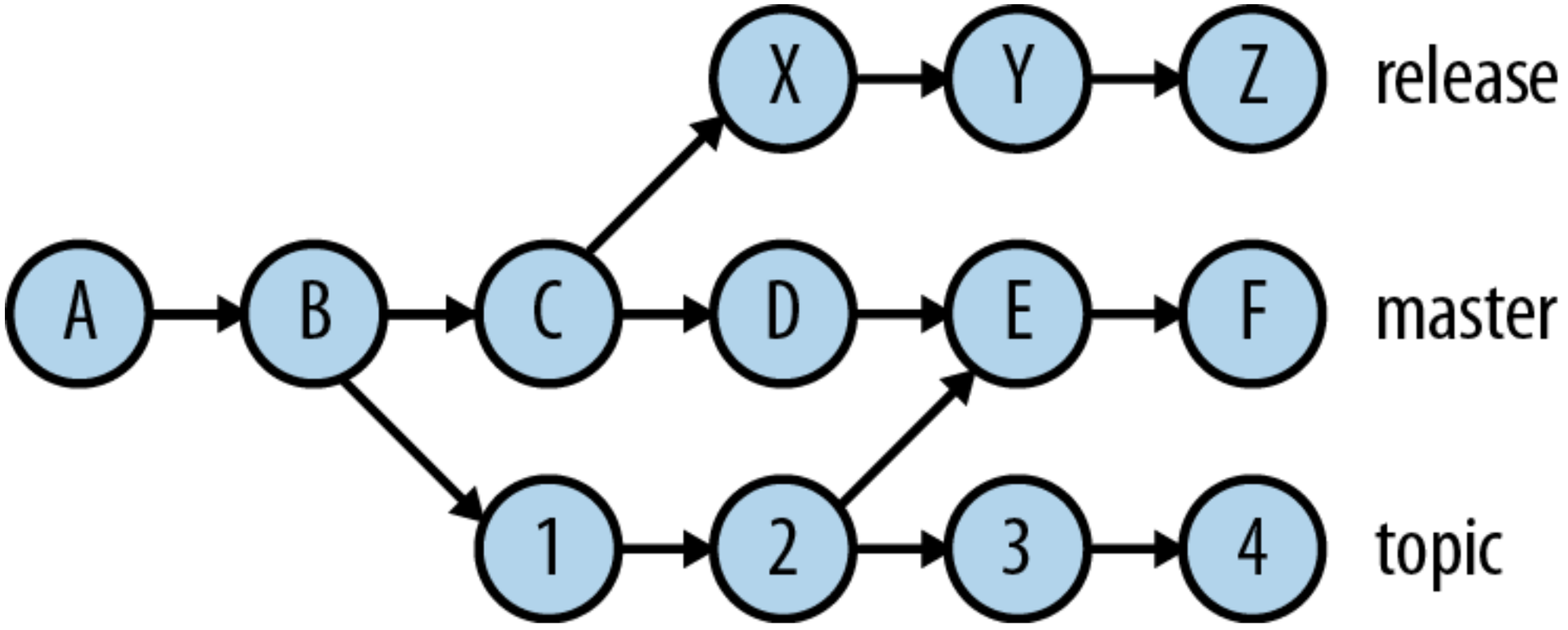
- 1991-2002 – ядро Linux существует в виде архивов и патчей
- 2002 – переход на проприетарную RVCS Bitkeeper
- 2005 – утеряно право бесплатного использования
- 7 апреля 2005 – первая версия git

Требования

- Скорость
- Поддержка нелинейной разработки (тысячи паралелльных веток)
- Полная распределённость
- Возможность работы с такими большими проектами как ядро Linux
- Unix-way

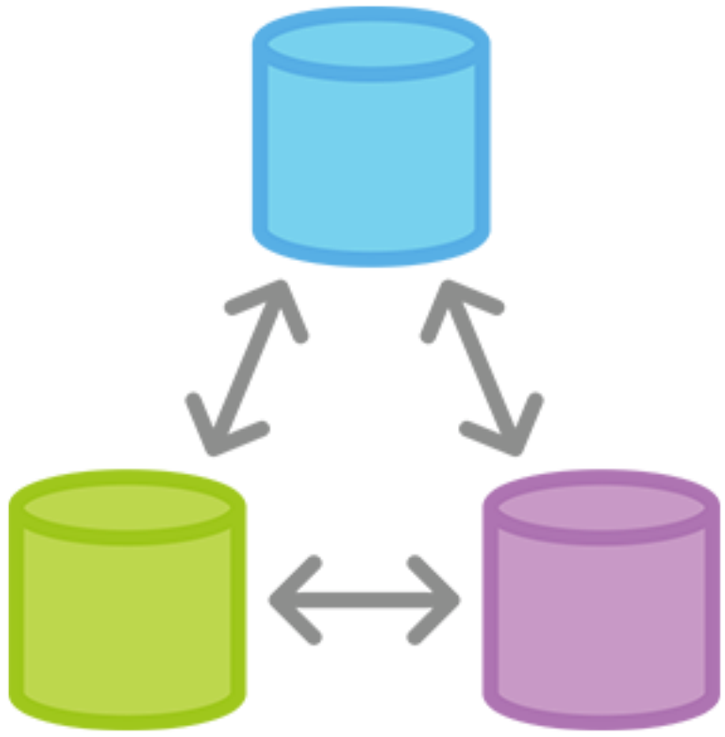
Git репозиторий

Весь репозиторий - граф



Git remotes

Git может работать с множеством репозиториев
Удалённые называются remotes



Работа с репозиториями (локально)

Commit – фиксация изменений – создание узла дерева

`git commit`

Merge – слияние – фиксация нового коммита с двумя (и более) предками

`git merge %branch% - слить %branch% с текущей веткой`

Работа с репозиториями (удалённо)

Синхронизация вверх
git push

Необходимо, чтобы репозиторий не имел новых коммитов
Нужно чтобы у коммитов были общие предки
Иначе нужно произвести синхронизацию.

Синхронизация вниз – pull
git pull = fetch + merge

Порядок работы

Всегда начинайте с git status

```
phantom@nzoth:~/tmp/git/git-sample$
phantom@nzoth:~/tmp/git/git-sample$
phantom@nzoth:~/tmp/git/git-sample$ git status
В ветке master

Заглавный коммит

Несопровождаемые файлы:
  (используйте "git add <file>..." чтобы включить то, что должно быть закреплено)

        README.md

нет изменений, добавленных в коммит, но существуют несопровождаемые файлы (используй
те "git add" чтобы добавить файлы для сопровождения)
phantom@nzoth:~/tmp/git/git-sample$
```

Важные команды/ действия

git add - добавить файлы в готовящийся commit
git commit - зафиксировать подготовленные файлы
git checkout %branch% - переключиться на ветку или коммит
git checkout -b %branch%

git pull %remote% %branch%
git push %remote% %branch%
git merge %branch% - слить ветку + разрешить конфликты

Конец