

如何使用工具提高PHP 项目的生产力

自我介绍

洪涛
@斯考吉恩

- PHP爱好者
- 目前的兴趣方向是移动领域
- 9年工作经验
- 5年团队管理经验
- 曾为多个大型互联网项目进行架构设计和开发工作
- 开源技术爱好者，喜欢研究各种开源项目并应用在工作中



- “我们要做一个新项目， 怎么开始比较好？”
- “目前项目结构太乱了， 怎么办？”
- “移动端的需求太复杂了， 有没有好的解决办法？”
- “测试部门报给我的问题在我电脑上是的啊！”
- 开发完的代码还没上线， 需求就已经变了怎么办？

三个挑战

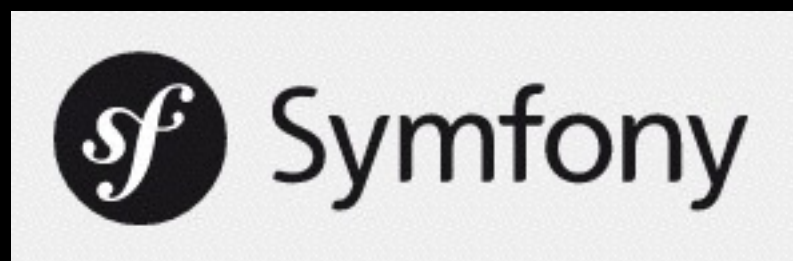
- 挑战一：如何进行代码和第三方程序的管理？
- 挑战二：移动互联网对现有网站架构的影响
- 挑战三：开发环境的最佳实践

挑战一：如何进行代码
和第三方程序的管理？

PHP项目结构进化史

- 最早，方法 + include/require
- 后来，类/命名空间 + __autoload()
- 现在，第三方软件包 + ?

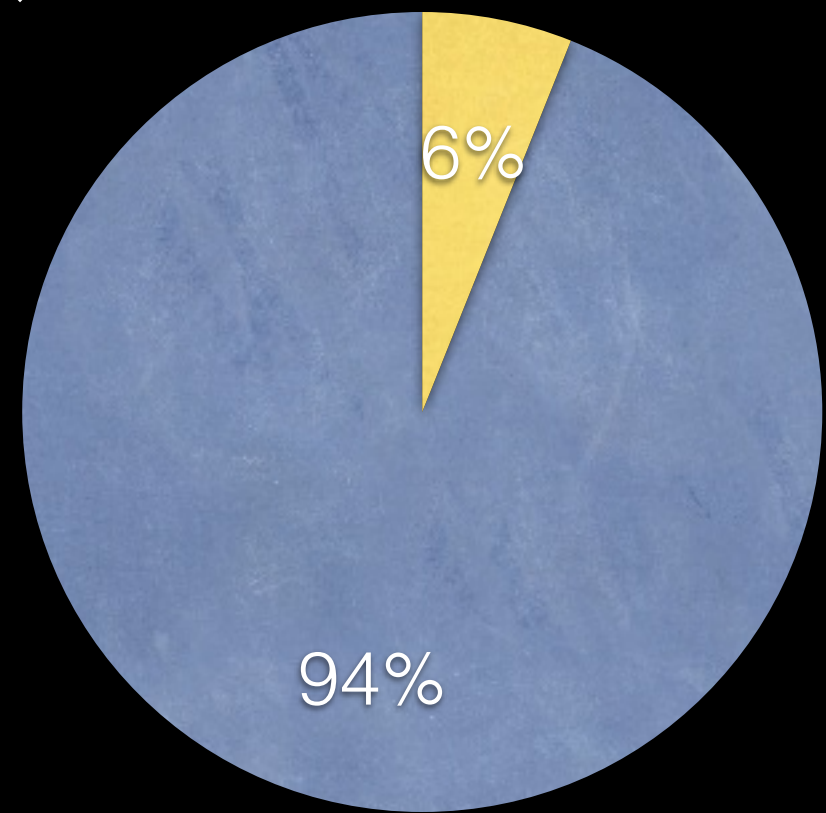
站在巨人的肩膀上



无数的开源项目.....

一个典型的中小型网站自身的代码和第三方代码行数对比

- 项目自身的PHP代码（含注释）
4万行
- 第三方PHP代码（含注释）
62万行



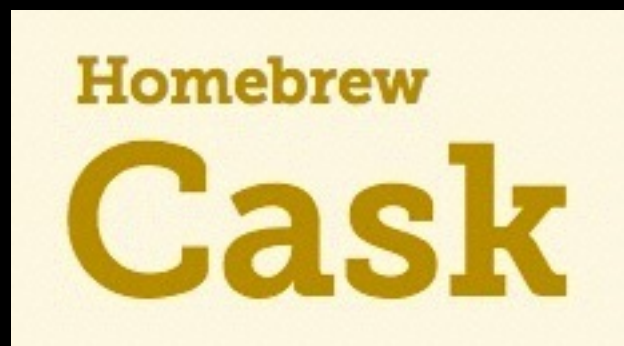
使用第三方程序作为基础 进行项目开发的一些难点

- 程序结构不同，无法整合
- 程序运行环境不同
- 版本升级困难

看看其他语言/平台



virtualenv
pip
setuptools



2个标准

- Namespace
- PSR-0

PHP下的包管理软件



Composer

- 方便地管理第三方软件包，能够实现自动升级
- 自动解决第三方软件之间的版本冲突
- 自动生成autoload脚本以加快项目载入速度
- 支持插件和hook功能

使用Composer之后的项目结构

- 项目自身的：
 - 业务逻辑代码
 - 配置信息
- composer.json文件

使用Composer之后的项目 准备流程

- 下载并安装Composer
- 在composer.json文件里定义该项目需要使用到的第三方程序
- 运行composer install命令，将这些第三方程序安装到项目目录内
- 按照composer提供的目录结构，开始项目开发

基本操作示意


- 例子：项目中需要安装一个新的第三程序：
Ladybug，ladybug能够方面的格式化调试信息
1. `composer require "raulfraile/ladybug": "dev-master"`
 2. `composer update`
 3. 在项目中即可直接调用 `ladybug_dump()` 命令

Packagist

Composer的
软件仓库:

packagist.org

[Create a new account](#) | [Login](#)



Packagist

The PHP package archivist.

[Submit Package](#)

Packagist is the main **Composer** repository. It aggregates all sorts of PHP packages that are installable with Composer.
[Browse packages](#) or [submit your own](#).

Getting Started

Define Your Dependencies

Put a file named `composer.json` at the root of your project, containing your project dependencies:

```
{
  "require": {
    "vendor/package": "1.3.2",
    "vendor/package2": "1.*",
    "vendor/package3": ">=2.0.3"
  }
}
```

Install Composer In Your Project

Run this in your command line:

```
curl -s http://getcomposer.org/installer | php
```

Or [download composer.phar](#) into your project root.

Publishing Packages

Define Your Package

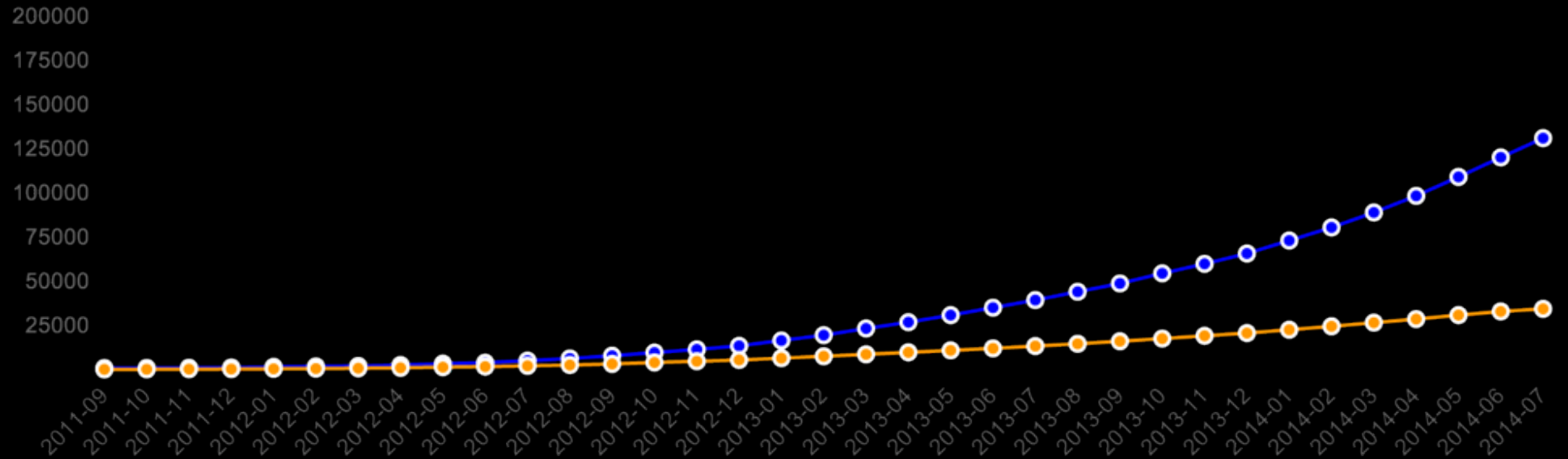
Put a file named `composer.json` at the root of your package, containing this information:

```
{
  "name": "your-vendor-name/package-name",
  "description": "A short description of
what your package does",
  "require": {
    "php": ">=5.3.0",
    "another-vendor/package": "1.*"
  }
}
```

This is the strictly minimal information you have to give.

For more details about package naming and the fields you can use to document your package better, see the [about](#) page.

Packagist



- 稳定运行3年
- 截至目前有3万4千个第三方软件的13万个不同的版本可供Composer直接使用
- 每天第三方软件被安装的总次数达到125万

Composer最佳实践

- 最新的开发版本(dev-master)并不一定是最好的。
- composer update时使用—prefer-dist或—prefer-source参数来提高更新速度
- composer本身也在不断升级，使用composer selfupdate命令进行更新
- composer不仅能管理PHP下第三方软件的，还能支持其他语言下第三方软件的管理

挑战二：移动互联网对 现有网站架构的影响

当我们谈论“移动互联网” 时，我们在谈论什么？

- 多维度化：
 - 多个终端，多个版本
- 信息交互的复杂化
- 开发的复杂化
 - “网站开发我太熟悉了，但API……”

难点

- 第三方授权，OAuth，权限管理
- REST/RPC
- 错误处理
- 多版本管理
- 格式处理及封装，Json/XML与CoreData/Java Class Mapping
- 如何协同接口开发和APP开发
- 文档

PHP下的API开发及管理工具

Apigility



Apigility介绍

Apigility是由Zend公司开发的基于Zend Framework的一款API开发框架

主要功能有：

- 支持Rest/RPC
- 支持权限管理/OAuth
- 有统一的错误处理和HTTP状态管理
- 能够单独使用也可以整合在现有的项目和框架中
- 支持在线调试/文档生成和查看
- 友好的管理界面

127.0.0.1:8888/apigility/

Swagger UI

127.0.0.1:8888/apigility/ui#/api/test_api_1/v1/rest-services?service=Post&view=settings

apigilityby zend framework

Settings

APIs

API Docs

Home / APIs / test_api_1 (v1) / REST Services

REST Services

Current Version: 1

Create New REST Service

Overview

REST Services

RPC Services

Authorization

User

Notification

Post

Photo

Car

Settings

Fields

Documentation

Source Code

General Settings

Route matches:
/post[:post_id]

Page Size
25

HTTP Allowed Entity Methods
GET, PATCH, PUT, DELETE

HTTP Allowed Collection Methods
GET, POST

REST Parameters

Content Negotiation

Service/Class/Route Names

API documentation

www.weimi.com/api/doc/

POST /ios/v1/man/unfollow 取消关注man(完成)

POST /ios/v1/man/user/feed/list 姑娘对man的列表(完成)

POST /ios/v1/man/users 获取对man产生过feed的用户列表(完成)

GET|POST /ios/v1/man/weibo_fans get man weibo female fans

Api Notification

GET /ios/v1/notification/get

get unread notification

Documentation

Sandbox

Input

Filters

read = Value -

type = Value -

do = Value -

offset = Value -

size = Value -

Try!

Headers

Key = Value -

New header

Content

Content set here will override the parameters that do not match the url

Content-Type = Value Set header

Replaces header if set

Request URL

GET /ios/v1/notification/get

Response Headers [Expand]

200 OK

Response Body [Raw]

```
{
  "error": 11003,
  "error_description": "请重新登录一下"
}
```

GET /ios/v1/notification/setread

set notification read

使用Apigility之后的API开发流程

- 下载安装Apigility
- 进入Apigility的管理界面，进行权限、方法、版本号、错误信息等等的一系列设置
- 把你的业务逻辑填充在Apigility生成出来的模板代码中
- 使用Apigility的调试工具进行测试和整合

挑战三：开发环境的最佳实践

复杂的开发环境?

- 为什么开发环境变得复杂了?
 - 需求变化更快，产品发布更加快速
 - 第三方软件变多，配置及各种依赖各种变得复杂
 - 程序员不再只是程序员，身兼多个职责，负责多个项目
 - 生产环境变得复杂了
- 开发者的环境变得复杂了
 - 移动办公、协作办公、结对编程、代码Review...

PHP开发环境进化史

- 最早，一键WAMP安装包+本地虚拟主机
- 后来，共享开发服务器 + 远程虚拟主机 + Mount
/var/www/scourgen/project1 =>
<http://project1.scourgen.dev.local>
- 现在，本地虚拟机/远程虚拟机，可是.....

虚拟机管理工具



Vagrant简介

- Vagrant不是虚拟机，而是虚拟机控制软件
- Vagrant帮助你搭建一个高效的虚拟机开发环境
- Vagrant可以适配各种虚拟机，例如VMware，Docker，甚至云主机，例如AWS
- Vagrant允许你通过各种系统构建软件和脚本，从无到有构建一个完整的软件运行环境
Chef, Puppet, Ansible, Shell.....

使用Vagrant后的开发环境 搭建过程

- 下载Vagrant配置文件，然后用命令行启动系统构建过程。
`#vagrant up`
- Vagrant自动帮你更新并设置好虚拟机内的开发环境
- Vagrant帮助你：
把代码目录映射到你本地
把虚拟机的80端口映射到你本地的8080端口
更新Host记录
- 用IDE打开项目开始开发
`http://your_project.dev/` => 指向虚拟机

系统构建过程脚本化的意义

- 快速构建系统
- 构建系统的过程属于软件开发的一部分
- 系统构建过程的标准化
- 是团队走向云计算和scale out的必经之路

Vagrant在开发过程中的应用

- 快速搭建开发环境
- DevOps，整合开发和运维流程，项目的开发和系统运维更加紧密结合
- 部署流程脚本化，使大规模部署应用变为可能

Vagrant支持

- IDE
 - PHPStorm
- 虚拟机
 - VirtualBox, VMWare, libvirt, KVM, Hyper-V, LXC/Docker

Vagrant入门项目介绍

- Box集合网站: vagrantbox.es
- 推荐两个用Vagrant搭建Wordpress环境的项目:
<https://github.com/chad-thompson/vagrantpress>
<https://github.com/Varying-Vagrant-Vagrants/VV>
- 在线配置并生成Vagrant配置文件的网站:
<https://puphpet.com/>
<http://getprotobox.com/>

Vagrant的最佳实践

- 很多Box通过vagrant直接下载速度很慢，建议用多线程下载工具下载到本地后用命令行直接注册
- 默认的源安装速度很慢，建议编辑脚本，换成国内的源
- 目录同步方式使用NFS可以提高文件同步的速度
- 目录映射的方向很重要
- Vagrant的几个实用的插件：
vagrant-hostmanager,vagrant-berkshelf,vagrant-omnibus

其他一些能够提高生产力的工具

Ladybug

```
<?php
$var = array(
    array(
        'name' => 'Raul',
        'age' => 29
    ),
    array(
        'name' => 'John',
        'age' => 27
    )
);

ladybug_dump($var)
```

array (2)

0 array (2)

name string(4) "Raul"

age int 29

1 array (2)

name string(4) "John"

age int 27



httpie

- 比curl好许多倍的http命令行调试工具

```
jakub@mac:~  
$ curl -i -X PUT -H 'Content-Type: application/json; charset=utf-8'  
-d '{"hello": "world"}' http://httpbin.org/put  
HTTP/1.1 200 OK  
Content-Type: application/json  
Date: Mon, 30 Jul 2012 09:40:20 GMT  
Server: gunicorn/0.13.4  
Content-Length: 474  
Connection: keep-alive  
{  
  "origin": "88.146.3.217",  
  "files": {},  
  "form": {},  
  "url": "http://httpbin.org/put",  
  "args": {},  
  "headers": {  
    "Content-Length": "474",  
    "Connection": "keep-alive",  
    "Accept": "application/json",  
    "User-Agent": "curl/7.24.0",  
    "Host": "httpbin.org",  
    "Content-Type": "application/json; charset=utf-8"  
  },  
  "json": {  
    "hello": "world"  
  },  
  "data": "{\\\"hello\\\": \\\"world\\\"}"  
}  
$  
  
jakub@mac:~  
$ http PUT httpbin.org/put hello=world  
HTTP/1.1 200 OK  
Content-Type: application/json  
Date: Mon, 30 Jul 2012 09:38:47 GMT  
Server: gunicorn/0.13.4  
Content-Length: 474  
Connection: keep-alive  
{  
  "args": {},  
  "data": "{\\\"hello\\\": \\\"world\\\"}",  
  "files": {},  
  "form": {},  
  "headers": {  
    "Accept": "application/json",  
    "Accept-Encoding": "identity, deflate, compress, gzip",  
    "Connection": "keep-alive",  
    "Content-Length": "18",  
    "Content-Type": "application/json; charset=utf-8",  
    "Host": "httpbin.org",  
    "User-Agent": "HTTPie/0.2.7dev"  
  },  
  "json": {  
    "hello": "world"  
  },  
  "origin": "88.146.3.217",  
  "url": "http://httpbin.org/put"  
}  
$
```

复杂开发环境下的团队协作 工具Slack



slack

但云也只是飘在各自头上的云

- 系统运维有云端的服务器
- PM有云端的任务管理工具
- 测试有云端的CI
- 程序员有云端的代码管理



BOSS

PM

DEV

QA

SA

OP

但云与云之间却是不连
通的

- PM说：我发现了一个工具很好，能不能和我们现有的系统整合？
- 系统监控工程师说：昨天晚上CPU占用率报警了，和昨天白天上的版本有没有关系？
- 测试工程师说：把3个月前发布的那个版本里的一个配置文件发给我。
- 运营说：昨天发的那批营销邮件未到达率很高，帮我看一下为什么。

Slack

- 一个可以帮你 执行脚本/压缩图片/检查代码/代码更新通知/服务器报警/发微博..... 的聊天室
- 一个带有聊天功能的团队协作软件

其他的解决方案?

- Github's Hubot + Third-party Adapters(Campfire,QQ,SKype.....)
- IRC server + script



Gulp



Grunt

- 前端开发利器
- 编译SASS/LESS+JS语法验证+HTML minify+CSS minify+压缩图片+刷新页面=零键完成

总结