



redhat.



About PHP Quality

PHP Tour – Nantes – May 2017

Talk by:

Remi Collet

Senior Software Engineer, Red Hat Inc.
PHP 7.2 Release Manager.

Summary



1. Introduction
2. Versions
3. Release cycle
4. Security management
5. PHP 7.2 roadmap
6. PHP QA (PHP, Fedora...)
7. Questions

Introduction

Remi Collet

- 1998 : PHP 3.0 user
- 2005 : Remi's RPM repository / LAMP
- 2006 : Fedora contributor (PHP stack)
- 2007 : Fedora PHP co-maintainer
- 2011 : PECL developer
- 2012 : Fedora / Red Hat PHP maintainer
- 2012 : PHP developer
- 2017 : PHP 7.2 Release Manager

=> <http://fr.linkedin.com/in/remicollet>

Remi's RPM Repository



- Cible : Fedora, RHEL, CentOS
- PHP 5.4, 5.5, 5.6, 7.0, 7.1, 7.2
- Paquets de base
 - Remplacement (php-*), 1 dépôt par version
- *Software Collections*
 - Installation en parallèle (php##-*)
- ~130 extensions
- Upstream de Fedora / RHEL / RHSCCL
=> <https://rpms.remirepo.net/>

Versions



Supported Versions



- PHP 5.6 security only (5.6.30)
 - Dec 2018
- PHP 7.0 stable, maintained (7.0.19)
 - Dec 2017 / Dec 2018
- PHP 7.1 stable, maintained (7.1.5)
 - Dec 2018 / Dec 2019
- Master – development (7.2.0-dev)

Enterprise distributions

- Red Hat Enterprise Linux 7
 - **php 5.4.16** + security (EOL Jun. 2024)
- Red Hat Software Collections
 - **php 5.6.5** + security fix (EOL Apr. 2018)
 - **php 7.0.10** + security fix (EOL Nov. 2019)
 - 7.1, 7.2...

Release Cycle



Versions cycle

- Major version every 3 years (7.x, 8.x...)
 - BC break if necessary (should be avoid)
- Minor version every year (7.1, 7.2...)
 - No BC break
- Patch version every month
 - No new feature

=> <https://wiki.php.net/rfc/releaseprocess>

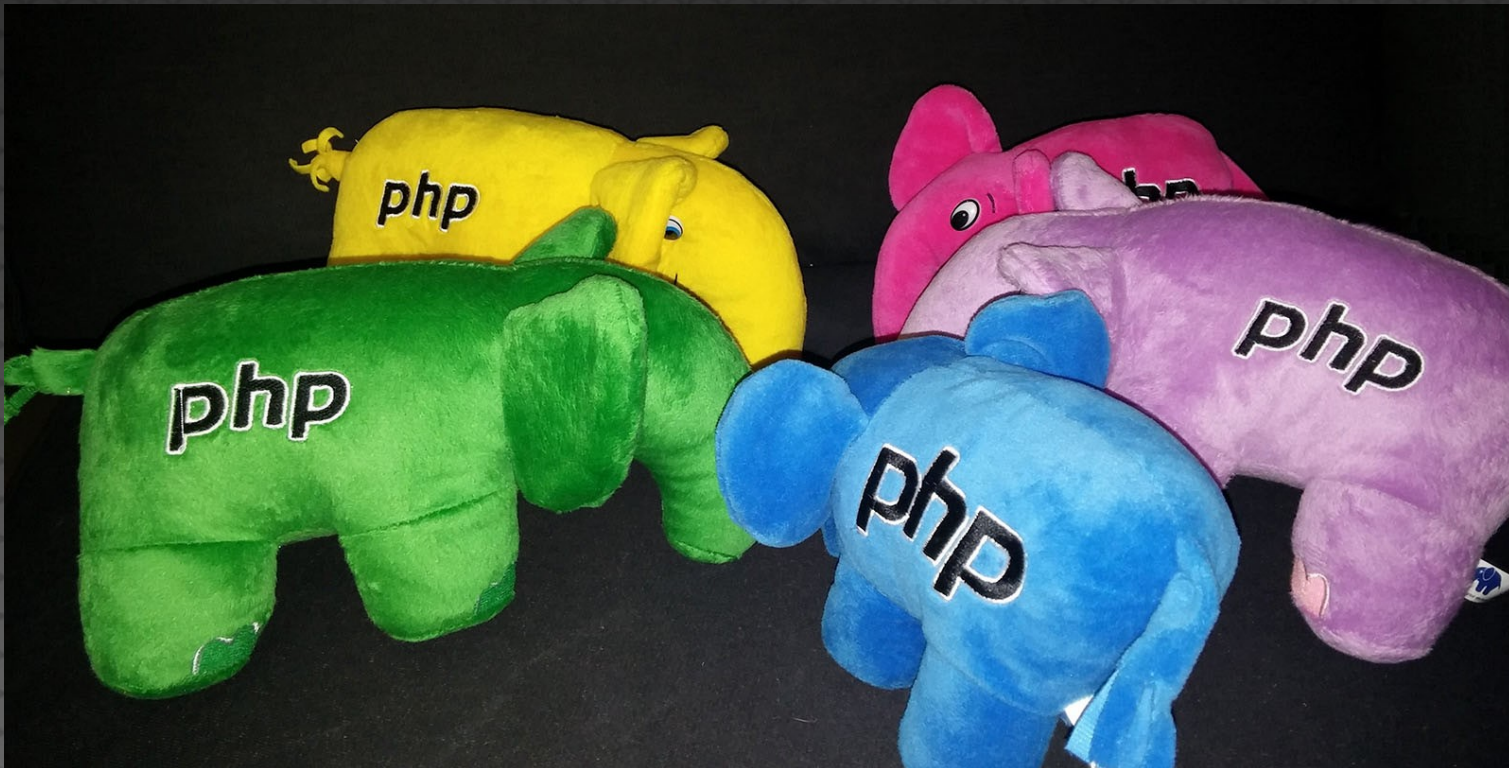
Release day

- Version tag created on Tuesday
- Version announced on Thursday

Release Candidate

- Standard rule
 - Latest Release Candidate = GA
- No change accepted...
- ...with some exceptions !

Security management



Classification

- **High severity**
 - third party to compromise any, or most installations of PHP
 - the execution of arbitrary code
 - disabling the system completely
 - access to any file a local PHP user can access.

=> <https://wiki.php.net/security>

Classification

- **Medium** severity
 - an extension that is not commonly used
 - a particular type of configuration that is used only in narrow specific circumstances
 - relies on old version of a third-party library being used
 - code, or patterns of code, that are known to be used infrequently
 - code that is very old, or extremely uncommon (and so is used infrequently)

Classification

- **Low** severity
 - This issue allows theoretical compromise of security, but practical attack is usually impossible or extremely hard due to common practices or limitations that are virtually always present or imposed.

Classification

- **NOT** a severity issue

- requires invocation of specific code, which may be valid but is obviously malicious
- requires invocation of functions with specific arguments, which may be valid but are obviously malicious
- requires specific actions to be performed on the server, which are not commonly performed, or are not commonly permissible for the user (uid) executing PHP
- requires privileges superior to that of the user (uid) executing PHP
- requires the use of debugging facilities - ex. xdebug, var_dump
- requires the use of settings not recommended for production - ex. error reporting to output
- requires the use of non-standard environment variables - ex. USE_ZEND_ALLOC
- requires the use of non-standard builds - ex. obscure embedded platform, not commonly used compiler
- requires the use of code or settings known to be insecure

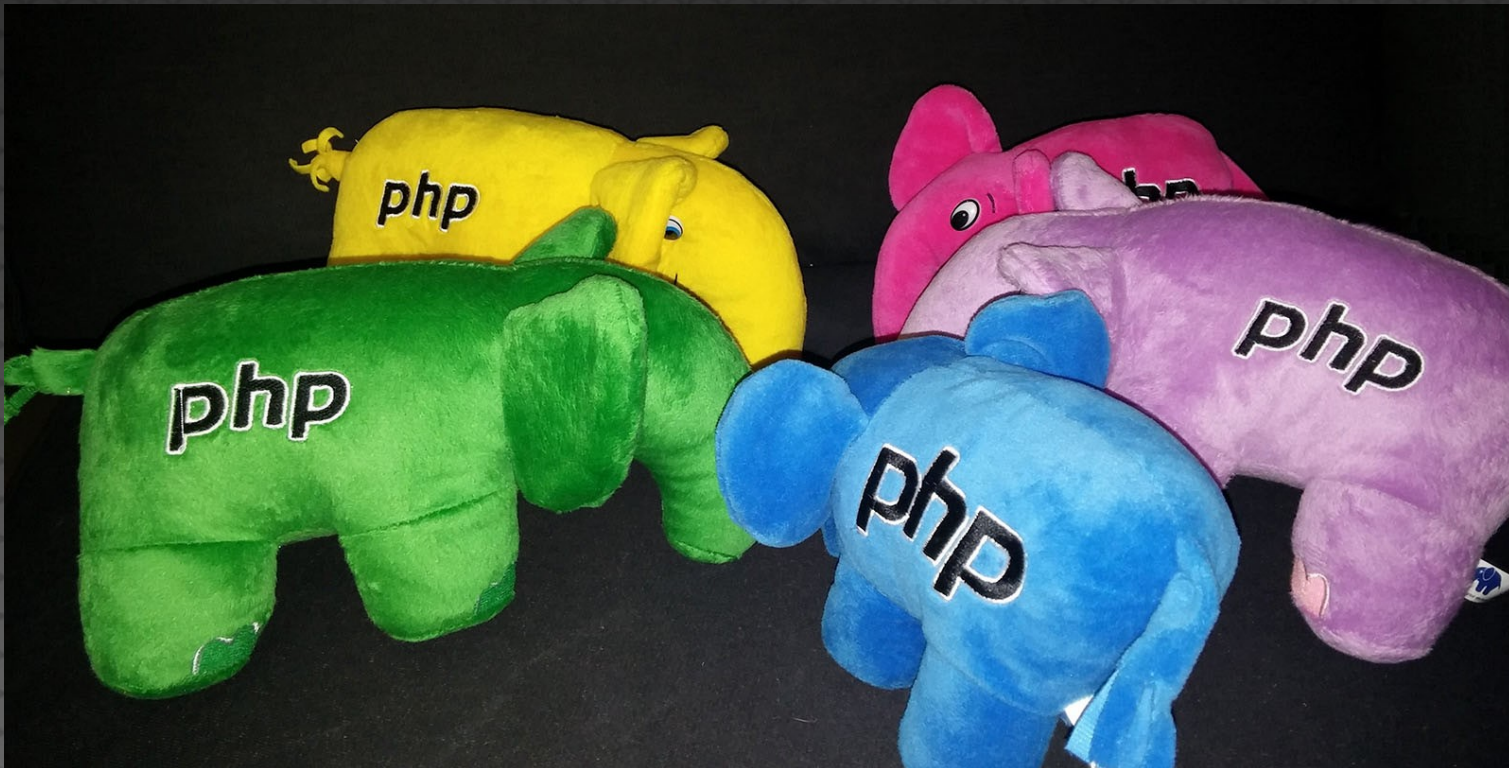
Classification

- **NOT** a severity issue
 - unserialize
 - memory_limit
 - etc

Management

- Reports
 - <https://bugs.php.net/>
 - security@php.net
- High and medium severity fixes are merged into a security repository and merged before the release is tagged.
- **Low severity fixes** are merged immediately after the fix is available and **handled like all regular bugs** are handled consequently. However, release managers may choose to pull those fixes into the RC branch after the branch is created, and also backport them into security-only release branch.

PHP 7.2



Roadmap

PHP 7.2 R.M.

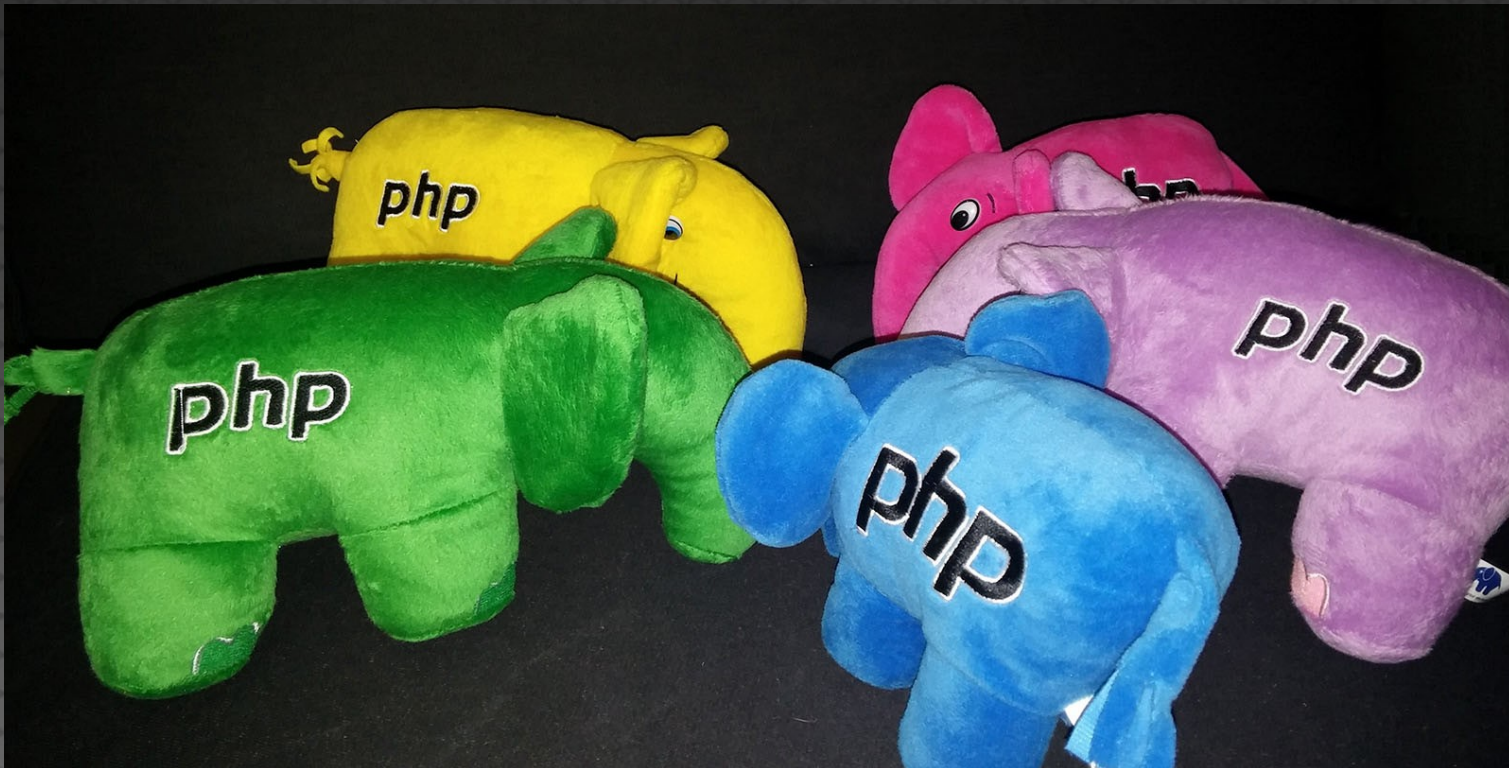
- Sara Golemon
- Remi Collet

PHP 7.2 roadmap

- Alpha (3)
 - Jun 8th, 22th, Jul 6th
- Beta (3) + PHP 7.2 branched
 - Jul 20th, Aug 3rd, 17th
- Release Candidate (6)
 - Aug 31th, Sep 14th, 28th
 - Oct 12th, 26th
 - Nov 9th + 7.2.0 branched
- 7.2.0 GA
 - Nov 30th

=> <https://wiki.php.net/todo/php72>

Quality Assurance



PHP test suite

- Change monitored by travis and appveyor
- Is PHP test suite really enough ?

Travis + Appveyor

- Used by PHP and various projects
- Fake distribution
- Build raised by changes in the project

Fedora QA

Koschei

- >500 packages monitored
- Build raised by changes in the dependencies stack
 - New PHP version
 - New system library version (icu, libxml...)
 - New PHP extension version (imagick, mongo...)
 - New PHP library version (Symfony, Twig...)
- Release Candidate version are used

Koschei



- Issue Found

- ICU 56.1 breaks php-zendframework-zend-i18n
- php-twig-1.23.0 breaks php-symfony-bridge-twig
- php-twig-1.23.0 breaks php-twig-extensions
- atoum methods inconsistency
- glibc 2.21.90 breaks Horde_Util
- regression in php-phpunit-PHPUnit-MockObjects 2.3.2
- git 2.4 breaks php-gitter
- rrdtool 1.5.x breaks php-pecl-rrd
- libxml 2.9.2 breaks atoum
- ...

7.0.18RC1 / 7.1.4/RC1 case

- ext-ast
- ext-yaml
- guzzle/guzzle
- guzzle/ringphp
- symfony

7.0.18RC1 + 7.1.4RC1 case

- Bug #74216 (reverted in 7.0.19RC1)
 - Misbehavior of "fsockopen" may introduce a security threat
 - `fsockopen("192.168.184.132:53", 80, ...`
- Bug #74429 (fixed in 7.0.19RC1)
 - Remote socket URI with unique persistence identifier broken
 - ```
$socket =
stream_socket_client('tcp://mysql:3306/root'
, $errorno, $errorstr, $timeout,
STREAM_CLIENT_CONNECT |
STREAM_CLIENT_PERSISTENT);
```



# PHP needs you

---

- Use RC versions
  - Windows build available
  - RPM available
- Start using 7.2.0-dev
  - RPM available
- Enable « nightly » in travis

# PHP 7.2 changes

---

- mcrypt dropped
- count()
  - only on countable
- preg\_math(\_all) => fix pending
  - empty string vs NULL
- stricter prototype check

# Feedback

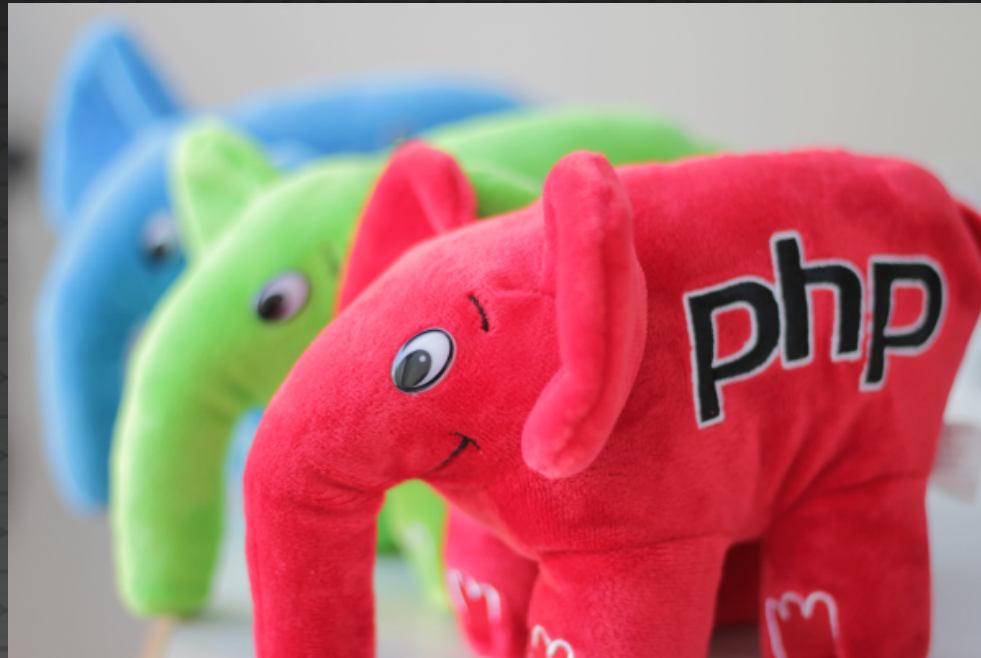
---

<https://joind.in/talk/d2aaf>





# Questions?



Contact:  
[remi@php.net](mailto:remi@php.net)