

Email marketing done the open source way - phpList

DevConf.CZ

January 24-26, 2020
Brno, Czech Republic



Mariana Balla – phpList Community Manager


- Community advocate
- Fedora Contributor
- Wikimedia volunteer
- Open Labs Hackerspace member





phpList


- Email marketing software
- 100% open source (AGPL v3)
- Initially launched in 2000
- Community driven





 phpList


 Dashboard


 Subscribers

 Campaigns

 Statistics


 System

 Config

 Update

RECENTLY VISITED

english

 Logout


Dashboard

★ Getting Started

Import subscribers

Start or continue a campaign


View statistics

 Last Campaign Results

Subject: phpList 3.4.0 released


9.729 Messages sent on 2 July 2019.

6.058 Viewed (62%), and 207 bounced (2%).

 help

- How to create a campaign
- How to use statistics

View Manual

 phpList community news

Tue, 18 Jun 2019

phpList 3.4.3 released: New domain statistics

Sat, 11 May 2019

phpList 3.4.2 released: Improved subscriber validation

Fri, 12 Apr 2019

phpList 3.4.1 released: REST API and Automatic Updater improvements

phpList (the software) & phpList Ltd.

phpList Ltd. is a UK registered company offering client support and developing phpList alongside with several community developers.

What is this session about?



3 different projects simultaneously

- phpList 3 (currently on 3.5.0)
- phpList 4 & REST API (WIP)
- CMS



Why considered a different development approach?

Releases were Feature-based not Time-based

- Once roadmap was fulfilled, there was a new release
- No consistent time frame for new releases

Feature based-why is this a problem?

- Inconsistent time frame
- Took longer for the community to get back & benefit from the features they wanted
- Community couldn't get involved further
- Testing phase took longer time because was internal

So??



What was considered on the way?

How Agile we want to be actually?

- Do we have the resources to support Agile?
- How quick can we adopt?
- How it would affect our way of doing things?

Considering options

- Every 6 months/ 3 months/ 1 month
- Monthly releases is the “most Agile” approach & would solve the problems raised

How it would affect our daily activity?

- Would we manage to meet our deadlines?
- Did our resources were enough to get things done?

How would the community react?

As of context: New releases need to be tested. Upgrading can lead to a ton of errors. Spending time troubleshooting upgrade/configuration problem would reduce the available testing time. So?

Introducing the Automatic Updater

Automating the upgrade process in order to be less time consuming & have low risk of errors.

Choosing our tools

In order not to introduce a new learning curve, we decided to to what we already have differently.

- Mantis bugtracker – roadmap
- Discourse – be in touch with the community
- Nextcloud – meeting notes
- Jitsi – daily stand up calls

Challenges faced

- Whats is really a “priority” and what it is not when you have a small team of devs.
- What happens when roadmap goals are not met?
- Managing backlog.
- How to manage community involvement?
 - People submitting PRs & joining discussions on GitHub VS people asking for all sorts of features.

Where we stand after 17 months?



Where we stand after 17 months?

- An open source software 17 times more updated
- Version 3.5.0 released on Jan 14th 2020
- Community submitting PRs on every release – more involved than ever
- Features get delivered faster to the community

Thank you :)

- ➔ twitter.com/phpList
- ➔ github.com/phpList/

- ➔ twitter.com/mariana_balla
- ➔ mariana@phplist.com