

CUNY Academic Commons - Feature #712

Plugin/theme request form

2011-04-27 08:09 AM - Boone Gorges

Status:	Resolved	Start date:	2011-04-27
Priority name:	Normal	Due date:	
Assignee:	Boone Gorges	% Done:	0%
Category name:	WordPress Plugins	Estimated time:	0.00 hour
Target version:	Not tracked		

Description

I would like to work up a more formalized procedure by which plugins and themes are added to the Commons.

I have a number of concerns related to this issue.

- Security. One lousy plugin/theme could potentially put all user data, and really the functioning of the whole site, at risk.
- Maintenance. When our users begin to use a plugin, and the developer of the plugin stops updating it, it means that the dev team is faced with a choice: find a way to get the plugin off the site (find a replacement, perhaps), or adopt and maintain the plugin ourselves. Both are extremely time-intensive, and we do not have an infinite amount of time.
- Support. Themes and plugins are inconsistent and often incompatible with each other. This means more support questions.
- Performance. This is especially a concern for plugins. If we have hundreds of plugins available, it makes it possible for users to activate many, many plugins at the same time, which can have an appreciable effect on our server resources if the site becomes popular.

There's not much of a process in place right now for adding stuff to the site. Only BuddyPress plugins are heavily vetted (because it affects the greatest number of users). I'm not so worried about these kinds of cases. On the other hand, in the case of non-BP plugins, I glance over the source code (looking for things like direct DB calls) and ensure that the plugin doesn't crash when activated. And for themes (especially ones that don't come from the WP repo) we run it through the Theme Check, which checks for malicious-looking code and some other technical things. These procedures are, to say the least, incomplete.

For that reason, I want to work up a document or two that will standardize the request and vetting process. I've listed some thoughts below about what such documents might look like. **Matt** and **Scott**, I've added you as watchers to see if you have any feedback regarding the requester questions. **Ray** and **Ron**, I've added you to see if you have any suggestions for the developer questions. This is totally off the top of my head, and is probably quite incomplete, so I'm anxious to hear some feedback. My comments are in brackets.

QUESTIONS FOR THE PERSON REQUESTING THE PLUGIN/THEME [generally this means Matt or a community facilitator]

- What's the name of the plugin/theme?
- [plugins] In a few words, what does it do?
- How is the plugin/theme different from what's already provided on the Commons?
- What's the potential impact? Who will use it?
- Who initially made the request? [sometimes Matt is passing along requests from members, sometimes suggesting something he found himself]
- Who is the plugin author?
- Say a little bit about the release history and popularity of the plugin. How many times has it been downloaded? When was it last updated? [The idea behind these last two questions is that I'm more likely to look favorably on plugins that are less likely to be abandoned]

QUESTIONS FOR THE REVIEWING DEVELOPER [generally me]

- Maintenance: Given the data above, how likely is it that the plugin is going to be abandoned by the developer?
- Security: Scan for XSS and DB vulnerabilities [How? Is there a better way than doing it by hand? Anyone have a script that greps for things like 'echo \$' and "INSERT "? I chatted a bit with Mark Jaquith in Phoenix and it sounds like he just has a list of things he searches for. Maybe we should invest in writing something.]
- Redundancy: If this plugin/theme does something that plugins/themes on the site already do, is there enough reason to put it on anyway? Maybe we should migrate users from the older plugin/theme to the new one? How hard would that be?
- Themes: What are the results of the Theme Check? Many of the 'required' warnings I don't care much about - they are wordpress.org nitpicks. But consistent calls to deprecated functions, for instance, could indicate something about the quality of the theme.

Thanks in advance for your help brainstorming, everyone.

History

#1 - 2011-04-27 02:53 PM - Boone Gorges

Daniel was kind enough to share the language he uses at the j-school: <http://tech.journalism.cuny.edu/request-wordpress-plugin/>

It's a nice start on coming up with some user-facing guidelines. It seems like a bit steep to request this much information from the end users of the Commons (though I'm willing to be persuaded otherwise) but it's certainly a helpful starting point.

#2 - 2011-04-27 09:50 PM - Matt Gold

Thanks for getting this discussion started, Boone. And thanks to Daniel for the J-School page (interesting to note, on the side, that they document new plugins and themes in targeted posts on their dev blog. That's something we should do, too, so that we have a way of telling users what plugins/themes were added when).

I very much appreciate the need for a system and for a better way of checking plugins. However, I'm a little concerned that the questions for the people making the requests set too high of a bar if our members will have to answer them (I'm fine with doing it if it's just me or Scott answering them). I want our members to feel like their feedback is encouraged and valued, and I worry that we'll be setting too many barriers in front of them. I'd prefer to have a relatively easy to use submission system that then draws them into a larger conversation in which we ferret out some of these details.

What do you think about having a relatively low barrier for a first submission, but then a second part of the process where these more rigorous questions get answered by our team in collaboration with the original submitter?

#3 - 2011-04-28 10:15 AM - Boone Gorges

I don't care **who** fills out the initial information. I agree that it's a bit much to ask this of most users. All I'm saying is that I would like to have this amount of information by the time that the request reaches me.

For what it's worth, I don't think we need to have any official channels at all for users to request plugins. The last thing we need is a torrent of requests. If people want **functionality**, that's great - but those requests already come through existing channels, and are filtered by members of the community and dev teams. I think it should stay that way.

I think that the j-school has a different kind of user base, so to some extent it may make sense for them to have a public method for plugin requests. I don't think we need to emulate that part of their process.

#4 - 2011-04-28 02:06 PM - Daniel Bachhuber

Little bit more input from me.

The policy I've put in place has largely only been useful because it now gives me the right to refuse to install plugins. Previously, students and faculty would ask for plugins to be installed. I'd get the sense they were frustrated when I would decline without detailed explanation as to why.

Moving forward, I think it would be wisest to have any public-facing material afford users the ability to request "functionality". IT can see whether that already exists, or go look for a plugin that meets code standards, etc. If nothing exists, then a decision could be made whether to build a plugin or not. In addition, we could pool requests as relevant and split up the work. I'm working on a Soundslides Embedding plugin with Jonathan Morgan at MSU that's a similar collaboration.

#5 - 2011-04-28 03:12 PM - Raymond Hoh

Re: Security

There are some scripts that help parse through the source code like [RIPS](#). I haven't really looked too much into the script or alternative ones. Could do some more research.

Re: Redundancy

If an older plugin is used and a suggestion is made to use a newer plugin, we'd have to notify users that are currently using the older plugin somehow. Let them know that the plugin in question is being phased out of CUNY Commons for the newer one. You could then use [Multisite Plugin Manager](#) to mass-deactivate the plugin in question. I can't think of anything better at the moment.

Re: Theme Check

I'm in agreement with you, Boone.

#6 - 2011-04-28 07:06 PM - Matt Gold

Boone Gorges wrote:

I don't care **who** fills out the initial information. I agree that it's a bit much to ask this of most users. All I'm saying is that I would like to have this amount of information by the time that the request reaches me.

For what it's worth, I don't think we need to have any official channels at all for users to request plugins. The last thing we need is a torrent of requests. If people want **functionality**, that's great - but those requests already come through existing channels, and are filtered by members of the community and dev teams. I think it should stay that way.

Great. I think we're agreed on this.

#7 - 2015-01-05 08:50 PM - Boone Gorges

- Category name set to WordPress Plugins

#8 - 2016-01-26 12:18 AM - Boone Gorges

- Status changed from Assigned to Resolved

We've had a form in use for a few years and it's worked well. Let's mark this resolved.