Drupal Contributions - A Pragmatic Approach to the Issue Queue
Who am I ??
AmyJune Hineline
She/Her/Hers
Open Source Community Ambassador at Kanopi Studios
@volkswagenchick
- Mother and Palliative Care Nurse
- Hobbies include: comics, geocaching, mushroom hunting, and volkswagening
Kanopi Studios

We create open source websites that make an impact.

www.kanopistudios.com
Who are you ??
Everyone!!
What are we going to talk about??

- Why contribute
- Benefits of contributing
- How to start contributing
- Keeping the contributors we already have
- Ways to foster new contributors
- Anatomy of an issue

If there is time after the demo, we’ll look at the new GitLab
Why contribute
Why Contribute

- If you depend on open source, open source depends on you
- Your contributions are valued
- You Get What You Give (and Then Some)
- Contributing makes you a more integrated part of the Drupal community
Come for the code,
stay for the community
Benefits of contributing
Benefits of contributing

More than warm and fuzzies

- Contributing makes you feel good
- Move the Drupal project closer to perfection
- Drupal street cred
- The more you contribute, the more you learn
Why individuals give back...

Build your resume

- Work on a team
  - Willingness to collaborate
  - Issues parallel working on tickets
- Write better code
  - Learn from other’s comments
  - Learn coding standards
Why Drupal shops should give back...

Why should agencies give back?

- Drupal depends on sharing
  - Good corporate citizenship
- Talent recruitment
  - Potential employees view profile pages
  - People grow to love the community and attending events
How to start contributing
Ways to contribute besides the issue queue

- Documentation
- External documentation guides
- Speaking at camps
- Translations
- Writing themes and modules
- Promote the platform
Ways to contribute in the issue queue...

- Report an issue
- Update issues
- Write patches
- Provide feedback
- Test patches

https://www.drupal.org/contributor-tasks
Keeping the contributors we already have
Why people leave...

- Burnout
- Work/Life balance
- Lack of recognition
- Lack of employer incentives
- Huge learning curves between releases
Fighting burnout

Find folks new roles

- Leadership
  - Maintain projects
  - Run initiatives

- Mentoring
  - Workshops
  - Speaking at events
Ways to foster new contributors
How do we recruit new contributors?

Mentor!!
Mentor!!
Mentor!!
Let’s talk issue queue!!
Find an issue...
The issue queue.... Start small and work your way up.

Patching can be daunting

Start with:

- Triage
- Tagging
- Test and review code
- Apply patches
There’s so many to choose from!?! 

Choose based on

- Skill ability
- Area of interest
- Actively maintained
Issues to steer clear of

As a beginner, pass on

- Issues with over 100 comments
- Issues that have changed status many times
- Projects that many open issues
- Multiple issues being closed with “won’t fix”
Report an issue...
What kind of issues are worth reporting?

All of them!

- UI errors
- Documentation
- Accessibility
- Bugs

You don’t have to know how to fix the issue to report it.
Search the queue...

Does it already exist?

- Update any changes
- Report similar findings
- Add steps to replicate
Anatomy of an issue...

A good ticket should provide correct metadata:

- Descriptive and concise title
- Category
- Priority
- Status
- Version
- Component
- Tags
Anatomy of an issue...

A good ticket should provide correct metadata:

- Title: [regression] Table Drag handles no longer respond to up/down arrow keys
- Project: Drupal core (3060)
- Category: Bug report
- Priority: Major
- Status: Needs review
- Version: 8.7.x-dev
- Component: javascript
- Assigned: Unassigned

Issue tags:
- accessibility, JavaScript, regression, Triaged for D8 major current state, Triaged D8 major, Vienna2017, badcamp 2018

Before adding tags read the issue tag guidelines. Do NOT use tags for adding random keywords or duplicating any other fields. Separate terms with a comma, not a space.
Anatomy of an issue...

Dreditor provides an issue summary template:

- Problem/motivation
  - Steps to reproduce
  - Expected results
  - Actual results
- Proposed resolution
  - Remaining tasks
  - Ui and API changes
  - Data model changes
Anatomy of an issue...

<h3 id="summary-problem-motivation">Problem/Motivation</h3>

<h3 id="summary-proposed-resolution">Proposed resolution</h3>

<h3 id="summary-remaining-tasks">Remaining tasks</h3>

<h3 id="summary-ui-changes">User interface changes</h3>

<h3 id="summary-api-changes">API changes</h3>

<h3 id="summary-data-model-changes">Data model changes</h3>

<h3 id="summary-release-notes">Release notes snippet</h3>
A good issue is worth its weight in latinum...
Write a patch...
Do you know how to solve the issue?

Write a patch:

- Download the repo
- Create a new branch
- Makes changes
- Compare the branches
- Upload the patch and write a comment
Review an issue...
Reviewed and tested by the community (RTBC)...

You don’t have to know code to test a patch!!

- Test the patch
  - Simplytest.me
  - Locally in git repo
- Provide useful information
- Attach screenshots
- Update the status
Reviewed and tested by the community (RTBC)…
Reviewed and tested by the community (RTBC)... Dreditor

<table>
<thead>
<tr>
<th>Comment</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>#29</td>
<td>interdiff-26-29.txt</td>
</tr>
<tr>
<td>Elaman</td>
<td>3003707-29.patch</td>
</tr>
</tbody>
</table>

8.x-3.x: PHP 5.5 & MySQL 5.5, D8.7 5 pass
Add test / retest
Reviewed and tested by the community (RTBC)... Dreditor
Live demo!!!
The Process..

- Find a problem
- Download the repo
- File an issue
- Create a new branch
- Make changes
- Create a patch
- Upload a patch
The move to GitLab
Why move the issue queue to GitLab?

The need to modernize

- Adopt a standard workflow that will be familiar to others outside our community
- Preserve those unique elements of how we collaborate
- Leverage an expert partner
Gitlab workflow...
Resources

- https://hbswk.hbs.edu/item/the-hidden-benefit-of-giving-back-to-open-source-software
- https://dri.es/contributing-back-to-drupal
Thank you!!
Issue and download repo

- Navigate to project page’s issue queue and create a new issue
- Fill out metadata
  - Category: Task (usually)
  - Priority: Normal or minor
  - Status: remain the same, Active
  - Select latest branch (usually)
  - Component: Make a good judgement call
  - Assigned: Yourself (If you are going to work on it in the next couple of days)
- Save
- Take note of node # from issue just created
- Navigate to project page and select the version control
  - Select the desired version and show
  - Copy git clone link
    - example: git clone --branch 7.x-1.x
    - https://git.drupal.org/project/i18n.git
    - cd i18n
Creating a Patch

- Create a new branch
  - `git checkout -b [issue node number]-[comment number]`
- Make changes to new branch and save
- Commit changes
  - `git add .`
  - `git commit -m "useful commit message"`
- Create a diff
  - `git diff [old branch] [new branch] > [project name]-[descrip]-[node number]-[comment number]`
  - Example: `sticky_toolbar-capitalize-name-2981572-4.patch`
Useful Terminal Commands

- **Basic terminal commands**
  - `cd` - change directory
  - `mkdr` - make a new directory or file
  - `pwd` - present working directory (where are you)
  - `open .` - opens file in Finder

- **Git commands**
  - `git clone`
  - `git branch`
  - List all branches
    - `git branch -a`
  - Change branches
    - `git branch checkout [name of branch]`
  - Create a new branch
    - `git branch checkout -b [name of branch]`
  - `git status`
  - `git add`
  - `git commit`