

Empowering Community Engagement Through Improving drupal.org

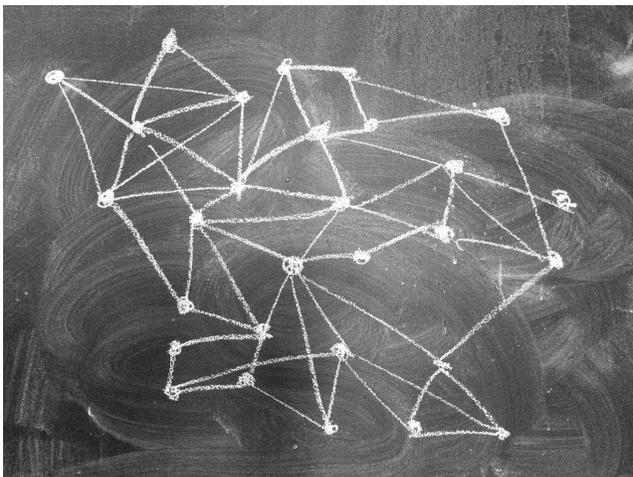
By Qasim Virjee (Principal, designguru.org)
July 22, 2013

As Drupal software is increasingly used by more people around the world to power a varied array of projects - from simple online publishing tools to complex web applications, defining the sense of community drupal.org facilitates becomes more relevant.

At base, the term 'community' can be applied to the meta group of people and organizations using Drupal around the world purely by the fact that they share this use in common. Taking the definition a step further, a more ideal 'community' would also imply a sense of fellowship between members which could result in common goals.

As the Drupal community grows, a greater importance must be placed on encouraging a strong sense of community whereby drupal.org can serve as a platform for achieving common goals of all members. This brief paper attempts to highlight some key areas of opportunity for encouraging such engagement.

Better leveraging the power of our network



Problem - Almost all professional firms and contractors using Drupal software are registered members of drupal.org yet the 'Marketplace' listing at <https://drupal.org/drupal-services> is limited and biased.

Suggestions:

To better represent the wealth of professional service vendors working with Drupal, Marketplace listings need to become better integrated with member profiles; when registering at drupal.org and/or filling in a profile, members

should be prompted to fill out a professional services description, which auto-lists in the Marketplace and collapses the anemic member directory at <https://drupal.org/profile> into a better functioning Marketplace/community listing (perhaps making commercial vendors a filter of the larger community list.)

There should be no distinction between 'Featured' and non-featured providers in the Marketplace; this confuses anyone coming to drupal.org looking to procure a vendor - instead, to establish vendor credibility we can offer tools such as ratings, testimonials/endorsements (as used on linkedin, for example) by previous customers to leave feedback on each vendor, and represent that information on their Marketplace listing.

Given that we work in a global context, and most Drupal professionals take on international client work, it seems redundant to only offer 'Browse by country' as a filter in the current Marketplace. Instead, filters would be more relevant if they related to skill-sets/services each vendor offer.

Problem - Lost opportunities for collaborative work.

Suggestions:

Based on how members tag their profile/marketplace-listing (listed services etc...), drupal.org should recommend other members to them; complimentary skill-sets need to be made obvious to members to encourage collaborative approaches to commercially working with Drupal.

Drupal vendors working together should be able to express their relationship publicly. Like the 'Connections' functionality on linkedin, vendors could ask each other to approve a simple flag (ideally with an annotation to explain how they work together) to display on each of their profiles.

Problem - Job Listings are too static.

Suggestions:

Though the main drupal.org job post/listing interface (<https://groups.drupal.org/jobs>) is hosted on drupal.org it is not integrated with marketplace functionality - so posters aren't automatically recommended vendors (through their service offering etc...) and vendors are not notified about the availability of postings unless they have been targeted to groups a member may belong to.

Tightening up the recommendation and notification cycle around procurement would encourage communication between posters and vendors and better humanize the procurement process.

As well, the current interface uses reverse-chronological listing; which buries older posts, even though they may not be filled. Instead, it would be worth considering alternate types of sorts which visually group listings by relevance to the person looking at them. Additional information at-a-glance should be offered through hover-tips to relate a summary of the position to the vendor looking through listings (often job descriptions are almost synonymous; who is looking for what is more important than the title they want to call the position which is available.)

Problem - Difficult to price procurement of Drupal vendors.

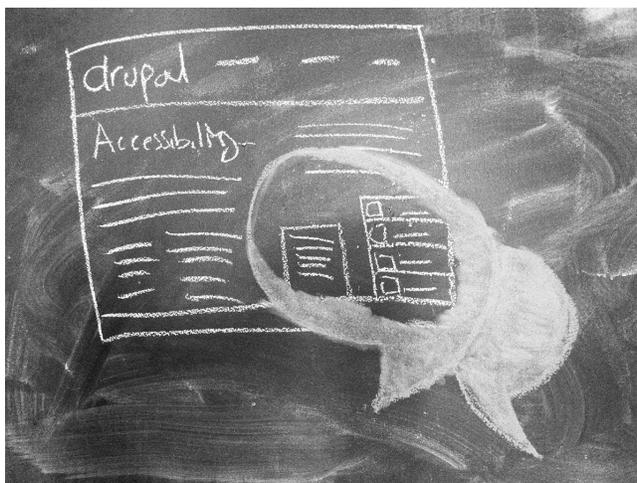
Suggestions:

Without standards for pricing varying tasks and quality of work, it will always take longer for procurement to sort through job listing responses etc... Vendors should easily be able to post case-studies formatted within set guidelines (to maintain congruency) with optional pricing; the case-studies should be available when viewing a vendor profile (individually and possibly through a simplified interface when looking at the marketplace.)

To allow customers of vendors to offer insightful feedback on the value of the service they experienced, a rating criteria should be offered to them so that the accuracy of pricing is maintained. For example, if vendor X posts case study Y with a guideline pricing/budget of \$100 and their client, Mister Z, rates the value of X's service as 80%, it is obvious to the case study reader that the project may have been better priced at \$80. As a large data set is auto-generated through submitted case studies and customer feedback, pricing recommendations can be automated as messages sent by drupal.org to vendors to suggest augmenting their price to better service customers (we could possibly have standards emerge as the aggregate pricing based on service per case study across the whole community.)

A simple pricing guideline can also help procurement; explaining how the community chooses to charge for their services. This could be tied to a set of standard methods which each vendor flags as their preferred pricing format. For example, 'Hourly rate' vs 'Monthly retainer' vs. 'All-inclusive per-project' etc...

Contextual Communication



Problem - Issue queues are support silos and IRC is abstract.

Suggestions:

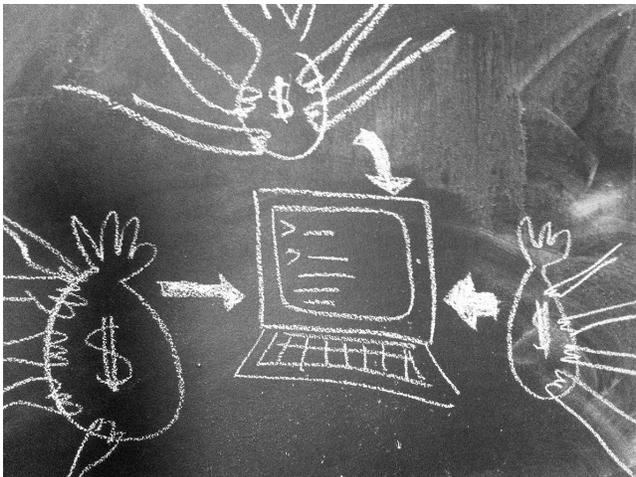
Currently live communication online between members of the Drupal community happens on Internet Relay Chat - in a myriad of channels (listed at <https://drupal.org/irc>) per specific topic of interest (e.g. accessibility vs commerce etc...) IRC is a great way for people to chat with each other but requires a separate client application; which non-technical community

members may not want to install; further, newer Internet users may not be familiar with command prompt conventions of IRC and thus feel inhibited from communicating with

the rest of the community using IRC. Besides from the learning curve of using IRC, it is further restrictive in not being contextual - because it requires a separate-from-browser application, communicating about information on drupal.org may feel discordant.

An integrated web-based chat facility on drupal.org (e.g. <https://drupal.org/project/drupalchat>) which displays to logged-in members of the community and features chat rooms per topic would be a huge improvement. If possible, this could just be a web client to an IRC server so that the existing conversations are not buried but instead brought right onto the drupal.org website. As members browse drupal.org and have immediate questions which may not warrant posting an issue queue, they should be able to simply type a chat message and be able to reference whatever they are looking at on their screen. A web-based chat system would also serve to better introduce members of the community to one-another as their on-site profiles would be linked to their chat IDs/handles (as opposed to the general anonymity of IRC handles.)

Financing Innovation



Problem - There is no direct way for members to sponsor development of modules which do not exist.

Suggestions:

There needs to be a marketplace by the community for the community - where module development can be proposed and answered by parties interested in working on specific modules. This would rely on better member profiling and need some automated notification whereby members can type in a time (or other resource) commitment of

availability to work (for the community) and the type of work they can offer.

Ideally drupal.org would notify members with related availability when a project has been proposed, and then voted for with a certain minimum weight by the community. In addition to being able to create a floated list of top-priority module development projects, an integrated crowd-funding system would ensure that adequate resources could be allocated to developing whatever modules the community needs. So, based on availability and remuneration needs, community members could actively work on module development and be paid for their time and effort.

Problem - When module development stalls, vendor credibility suffers.

Suggestions:

Many Drupal modules have been created to integrate Drupal software/websites with 3rd party services (such as [Mailchimp](#) for email newsletters or [Kaltura](#) for video.) Some of these are actually sponsored or even published by the 3rd party service but the majority are created out of needs specific client projects present to developers. A common phenomena seems to occur on drupal.org where a lot of work is put into creating an integration module and then its developer will not have adequate time or will to support its development; leaving modules out-dated as their integrated service adds features etc in time.

Often in these cases the 3rd party service vendor isn't aware that the lack of support for their Drupal integration is causing drop-off to their service use within the Drupal community, or unsure as to what extent that drop-off or dissatisfaction has reached. Ultimately drupal.org relies on issue queues per module to provide forum for discussing Problems of all kind with modules. As issue queues grow and developers become weary of supporting modules they've created out of lack of time/money to do so, it would be helpful if they can flag a module for additional support; this could be facilitated as a formal request to the integrated service (sent on behalf of drupal.org - lending brand equity and stature) and/or a request for support to the community.

Again, integrating a crowd-funding mechanism with drupal.org could allow not only new modules to be sponsored by the community but existing modules - so when a developer can't afford to work on their module, the community can help buy the time required for them to fulfill issue queue requests/feature-advancement etc...