

Tutorial

From Static and Stale to Dynamic and Collaborative: The Drupal Difference

Editor's Note: This paper is adapted from a presentation given at the 2010 LITA Forum.

In 2009, the University Library of the University of California, Santa Cruz, moved from a static, Dreamweaver- and HTML-created website to an entirely new database-driven website using the open-source content management system (CMS) Drupal. This article will describe the interdisciplinary approach the project team took for this large-scale transition process, with a focus on user testing, information architecture planning, user analytics, data gathering, and change management. We examine new approaches implemented for group-authoring of resources and the challenges presented by collaboration and crowdsourcing in an academic environment. We also discuss the impact on librarians and staff changing to this new paradigm of website design and development and the training support provided. We present our process for testing, staging, and publishing new content and describe the modules used to build dynamic subject- and course-guide displays. Finally, we provide a list of resources and modules for beginning and intermediate Drupal users.

Ann Hubble (ahubble@ucsc.edu) is Science Librarian, **Deborah A. Murphy** (damurphy@ucsc.edu) is Emerging Technologies Librarian, and **Susan Chesley Perry** (chesley@ucsc.edu) is Head of Digital Initiatives, University of California, Santa Cruz.

Why Change was Needed

Our old library website was created using static HTML and its organizational structure evolved to mirror the administrative structure of the library. The vocabulary we used was very library-centric and, though useful to library staff, could be confusing to patrons. Like many larger, older websites, we had accumulated a number of redundant and defunct pages. Many of these pages had not been updated for years, had inconsistent naming conventions, or outdated page design.

The catalyst for updating our web presence was predicated on several things. With more than one million visits per year and more than two million page views, our old servers were no longer able to handle this load, and we were about to begin a major project to replace our server hardware. In addition, we anticipated participating in an upcoming transition to a new campuswide website template.

We saw this moment of change as an opportunity to revitalize the library website's entire structure and reorganize it with a more user-centric approach to the menus and vocabulary. To do this, we decided to move away from Dreamweaver and the static HTML approach to web design and instead choose a CMS that would provide a more flexible and innovative interface.

Choosing Drupal

We had done research on commercial and open-source solutions and were leaning toward Drupal as our CMS. Many academic departments at our campus were going through a similar process of website redesign and had already explored the CMS options and had chosen Drupal. This helped move us toward choosing Drupal and taking advantage of a growing developer community on campus. Two of the

largest units on campus both chose Drupal as their CMS and have since been great partners for collaboration and peer support.

Drupal is a free, open-source CMS (or content management framework) written in PHP with a MySQL database backing it up. It is a small application of core modules with thousands of add-on modules available to increase functionality. Drupal also has a very strong developer community and has been adopted by a growing number of libraries. We have found it to be very open and fluid, which is both a blessing and curse. For any one problem there can be dozens of differing solutions and modules to resolve it.

The Transition Team

The library created a core Website Implementation Team consisting of a librarian project manager/developer, a web designer from the IT department, and two librarian developers. The core team was supported by a server administrator and an IT analyst. The IT staff supported the technical aspects of Drupal installation, backup, and maintenance. The librarian developers planned the content migration and managed the user interface design, layout, content, scope, and architecture. They needed to know the basics of how Drupal works and needed to have much more access to the inner workings of Drupal (e.g., modules, user permissions, etc.) than staff. The librarians also would train library staff, so needed to be able to teach and develop documentation and tailor instruction to specific staff needs. Everyone who participated in the implementation team had many other competing responsibilities. The librarian developers had other projects and traditional duties such as collection development and reference services, so learning Drupal and creating this new website was a part-time project and had to be integrated into existing workloads.

Transition Planning

With the goal of making our new site user-centered, we wanted to make data-driven decisions about design rather than what had ultimately devolved into the practice of decisions based on politics and committee negotiations. To that end, we took several approaches to gathering user data. We inventoried our current site and gathered usage statistics based on website analytics. We met with small campus focus groups who answered questions about library site searching. We created personas for user categories based on profiles of average users (e.g. first-year students, graduate students, faculty, community users, etc.). Based on this data, we drafted web interface wireframes and began user testing.

Drupal implementation also included developing a safe and effective means of moving from a testing environment to a final, public production site. This deployment process is a crucial component of ensuring that we could both test new features and still provide a stable environment for our users. After extensive discussions and revisions we developed a process to experiment with new modules and themes in a way that does not overwrite our existing public site. The deployment process goes from development to staging to production. It is critical to be able to determine that a new module or update will not negatively affect the database. The process we follow from our sandbox site to our production site is described in more detail in appendix A.

Transition Team Training

We had three types of Drupal users within the library: system administrators, developers, and staff (the primary content editors); each group had its own training needs. The library project manager, web designer, and IT systems administrators benefited

initially from the increasing interest in Drupal at the campus level. They attended a two-day intensive Drupal training course from a company called Lullabot, which provided an in-depth technical foundation for our initial Drupal installation.

This level of technical training and content was not appropriate for the other librarian developers on our team. However, a more detailed, mid-level training would have benefited the librarian developers and moved the project forward at a faster pace. These librarian developers learned using a combination of resources, including free online content that covers core Drupal skills, combined with a few carefully chosen professional in-person consultations and online training packages and books. Drupal is not a static environment, so after the initial training there was still a need for regular updates and refresher courses. Our transition team joined the Drupal4Lib discussion list and consulted with library colleagues using Drupal in the northern California area. DrupalCon conferences as well as online users groups were excellent places not only to learn but also to make contact with vendors and other developers. Several of these resources are listed in the accompanying bibliography in appendix B.

Staff Training

By far our largest group of library Drupal users was the fifty-plus library staff content contributors who were faced with learning a new approach to web development. Drupal's successful implementation was ultimately dependent on ensuring library staff would be able to create, edit, and manage thousands of library webpages using this new CMS.

This was a change for everyone in the library, not just a few. The new website meant leaving behind the comfort of routines created over the years, elaborate designs that had been developed, and various idiosyncratic

and often eccentric organizational structures that were no longer meaningful. The previous website had accumulated pages that were a bit more freewheeling in design with a lack of consistent navigation and "look and feel." Adding another layer of complexity, our website changeover took place during a period of great organizational change and a severe budget crisis. Surprisingly, what seemed at first a major drawback was actually somewhat helpful. With fewer people spread thinner and doing more work, there was less need to feel in control of individual empires, leading to more cooperation during the changeover.

Staff learning styles vary, and no one approach to Drupal training will work for everyone, so we brought many of the lessons we have learned in our bibliographic instruction sessions to our staff training. For example, we focused training on repetition, reassurance, and patience, ensuring it was an active process with hands-on participation as well as a lecture or demonstration. We provided ample time for questions and invited staff to bring their own projects to work on during training sessions. Though some staff only needed to learn a few applications within Drupal to perform their jobs, most needed specialized instruction to do some departmental-specific task or action that now had a very different interface. We supplemented our large group by drop-in training sessions with specialized departmental sessions, custom-made documentation, individual hands-on training, e-mail updates on system changes, and regular presentations of new system features. Not everyone will become a "born again" Drupalista, but everyone should at least feel that they can get their work done using Drupal.

Drupal has also meant changes not only in the way content is added to the website, but also in how we handle revisions and updates. In the past, we had a very siloed

permissions environment that limited editing to only those given specific permissions. We now have role-based ownership where everyone can edit everything so that we did not have to keep up a detailed list of who does what. Initial concern that someone could write over or accidentally delete pages was somewhat remedied by the Drupal Revisions History feature, which assists with version control. There have been a few pages where ownership is an issue, and we are still in the process of developing a system to ensure that pages are updated when there is no specific individual linked to a page.

Dynamic Displays: Article Databases and Subject Guides

As part of the move to Drupal, we wanted to take advantage of the new environment to redesign some of the more specialized portions of our site. In particular, we hoped that Drupal's dynamic displays would help us to keep our site more current with less effort. With this in mind, we chose to focus on two of the most heavily used resources: our list of article databases and our library subject guides. We planned to transform these static, high-maintenance HTML pages into dynamic, easily maintained, and easily generated resources.

We used a number of Drupal modules to develop the library's website, and these are described in more detail in appendix C. To redesign our list of article databases and our library subject guides, we relied heavily on three important modules: CCK (Content Construction Kit), Views, and Taxonomy. The interaction of these three modules is key to building dynamically created webpages. Once these modules are configured, information is input just once. Drupal does the work of pulling the right information from each resource to create dynamic displays. If information, such as a URL



Figure 1. Dynamic Display: Article Databases and Research Tools

for a particular database changes, it's updated in a central record and immediately reflected in displays throughout the site. Not only is it less work to update information, but we also can provide resources in more varied combinations and make them more findable for our users.

Figure 1 shows how the dynamically created "Article Databases and Research Tools" list appears to a user browsing our website. The default display lists these resources in alphabetical order. The user can display the same group of records sorted by other criteria just by clicking on the appropriate tab. If the "by Subject" tab is selected, the resources are displayed under subject headings. Selecting the "by Type" tab lists the resources by resource types, such as dictionaries and encyclopedias, citation style guides, etc.

Our subject guides are also created using the same components used to build the "Article Databases and Research Tools" lists. Figure 2 shows a portion of one of our subject guides. Like the previous example, this portion of the guide is created dynamically, displaying only records

appropriate for this particular subject. Each tab can also be customized to display whichever records pertain to this subject area.

How Our Dynamic Displays Were Built

CCK (Content Construction Kit) Module

Content used in both the "Article Databases and Research Tools" and "Subject Guides" displays is held within a special record type we created using the Content Construction Kit (CCK) module. We called this special record, or content type, Online Resource. We defined fields within the Online Resource record to hold information about individual resources we want to either display on our website or keep track of internally. The fields we defined include the resource name, URL (sometimes multiple URLs), description, type of information (article database, dictionary, encyclopedia, etc.), and subject discipline. Figure 3 shows what a portion of the Online Resource record

and list of defined fields looks like behind the scenes to the librarian web developer.

Some fields within the Online Resource record rely on a little further customization. The “Type of Information” field is defined via an Allowed values list. Figure 4 shows a portion of the values list we have defined for this particular field.

The “Subject, discipline, topic” field (figure 3) incorporates a taxonomy list that we first created using the Taxonomy module. This taxonomy vocabulary allows us to later sort the resources dynamically in both the “Article Databases and Research Tools” (figure 1) and “Subject Guides” displays (figure 2).

Taxonomy Module

Figure 5 shows the list of subject terms we created using the Taxonomy module. Terms are easily added, edited, and organized via this taxonomy display, available only to the web developers.

Views Module—Putting it all together

To define how the Online Resource records are displayed to the user (figure 1), we use the Views module. Views allow us to define, sort, and filter these records for display. Figure 6 shows what the “Article Databases and Research Tools” view of figure 1 looks like to the web developer. Notice that “A–Z,” “Subjects,” and “By Type” are listed in a box on the left side of the page. Each of these tabs corresponds to a tab on the page that displays to the user. In this case, “A–Z” is bold and is the active tab currently being defined for this display. Display settings such as the record type used, number of records to display per page, specific fields to display, type of sorting, and URL path for the webpage are defined here.

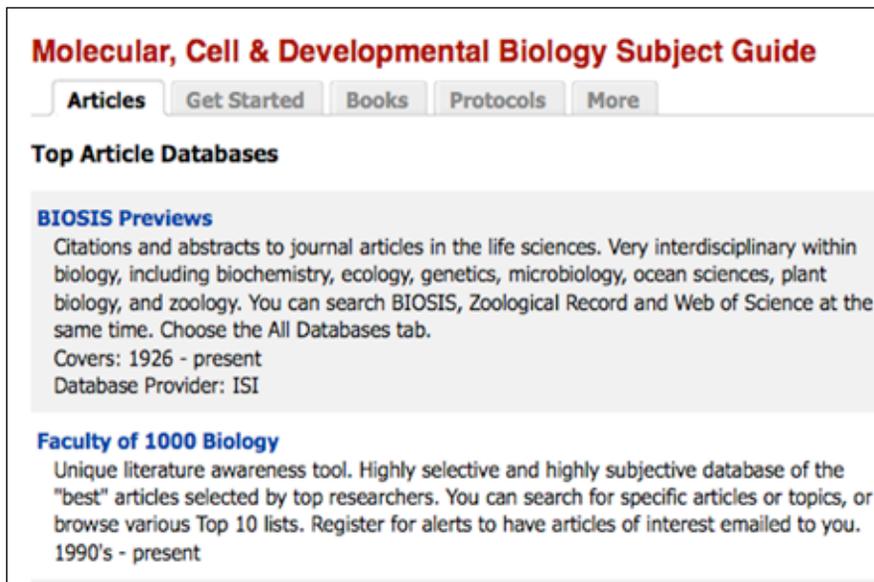


Figure 2. Dynamic Display: Dynamic Portion of a Subject Guide

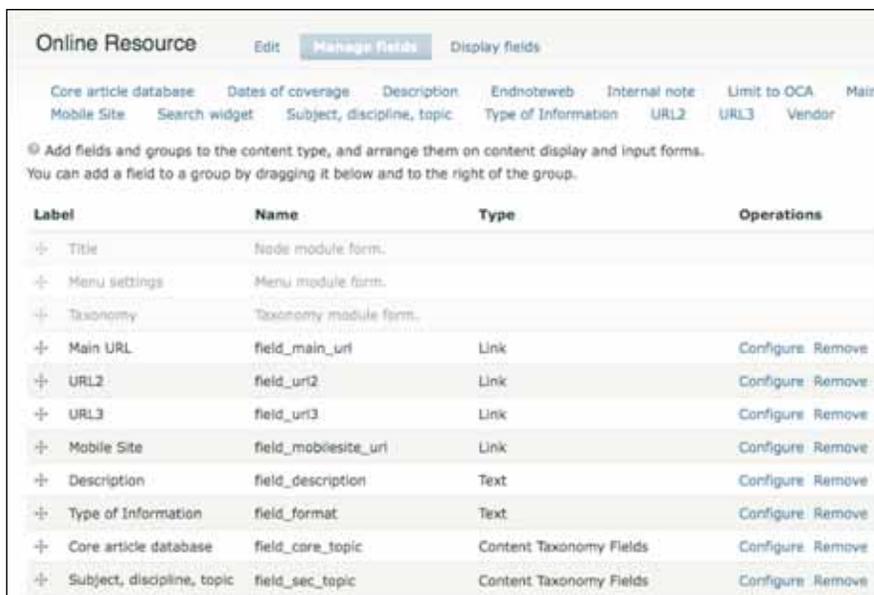


Figure 3. CCK Module: Online Resource Record: Manage Fields

Lessons Learned

Learning Drupal Takes Time

Our Implementation Team was comprised predominantly of librarians

and not programmers. We found that Drupal was very different from anything we had used before and had a very steep learning curve. If we could start over, we would have invested much more time in

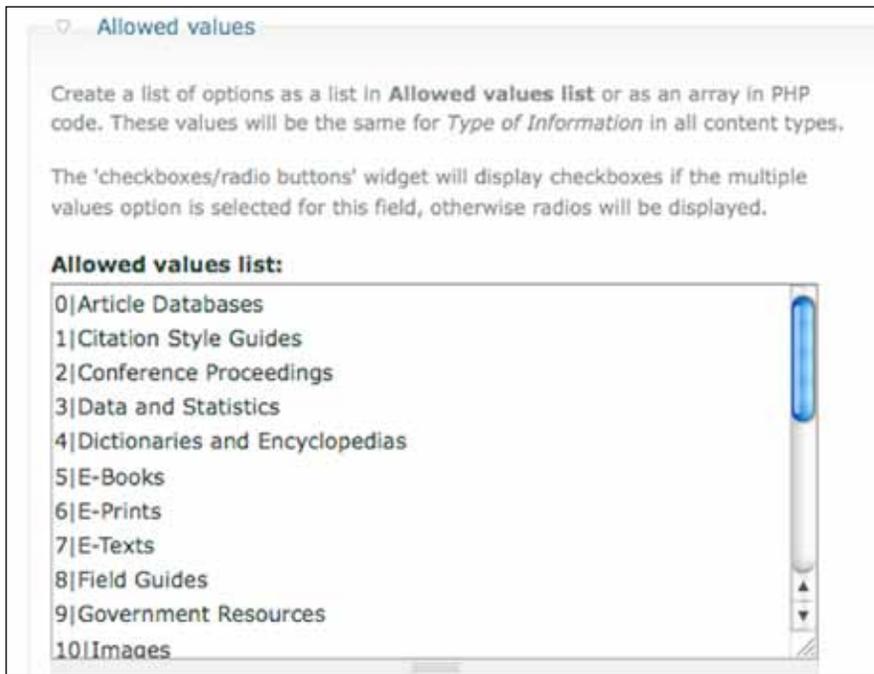


Figure 4. CCK Module: Allowed Values List (Type of Information)



Figure 5. Taxonomy Module: Subjects List

learning Drupal basics and getting a better grasp of how Drupal works as a CMS. The architecture and database-driven paradigm of our new Drupal site is a significantly different environment from our previous website's HTML-designed pages and directory-and-folder organization. Of particular importance for our site were three core modules: CCK, Views, and Taxonomy. Becoming proficient with these modules was a challenge, and we can't emphasize enough the importance of good, basic training on their use.

Start Small: Identify Small Parts to Bring Over

Initially, the thought of moving our old website to Drupal seemed insurmountable. Bringing over static HTML pages was straightforward, but portions of the website (such as converting our database of online resources) took more intensive planning. The entire process became more manageable when we divided up the site and focused on Drupalizing small parts at a time. This way we could focus on learning enough Drupal to make these portions of the site work without being overwhelmed.

Project Management Software: Document & Share What You've Done

If we were to transition an entire website again we would recommend using some type of project management software before starting. None of the Implementation Team worked on this site full time. This project was added to our other full-time workload providing reference services, collection planning, teaching, digital projects, etc. During our project we tried several free products but were not satisfied with any of them. We felt that finding the right project management package could have made the website transition process much

easier. Documenting and sharing how we created elements of the site helped us replicate complex components and allowed us to collaborate more easily on various projects.

Test, Test, Test

Testing the website as we developed it was a crucial component of our work. Modules also can interact with other modules in unpredictable ways, so we ultimately found that loading new modules on our sandbox site, a mirror of the library website, was a crucial step in determining compatibility as well as functionality with our existing site (appendix A). It's essential to practice using a live site without bringing the real production website down.

Focus on Essential Modules: CCK, Views, Taxonomy Images, WYSIWYG editors

Drupal comes with a set of core modules plus an ever-increasing number of specialized contributed modules. Finding and installing the right contributed module that fits a particular need can sometimes be difficult. There are often myriad modules that can solve a problem. It takes time to find and test each one to see if it will actually function as needed, and not all modules work well with one another. Focusing on the essential Drupal core modules plus CCK, Views, and Taxonomy will help reduce unnecessary development frustrations.

Staff are Important

Though we created many personas for faculty, students, and community users, we should have created

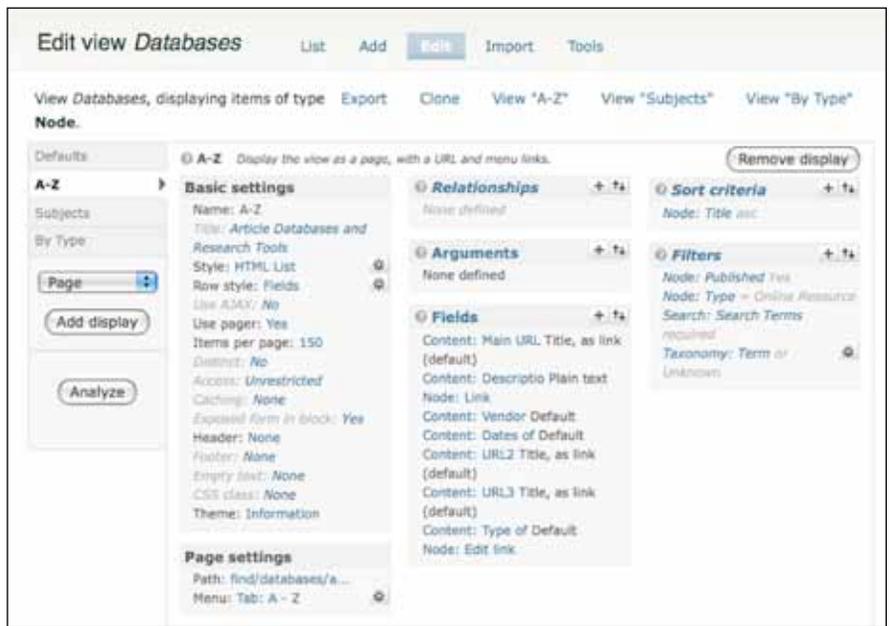


Figure 6. Views Module: Article Databases and Research Tools View

personas for librarians and content contributors and done more usability testing for non-developers. We found that training and teaching library staff the architecture and database-driven paradigm of the new Drupal culture has been a challenge and we still have varying levels of buy-in.

Conclusion

We now have a consistent look and feel to our site, though there are still many things yet to do. Now that we are more comfortable using Drupal, we can focus on creating more dynamic content, such as staff lists, adding sidebars to pages, and so on. Increasing the number of dynamically created pages will mean a more up-to-date site in general. Though group authoring within the library is still a challenge, we continue to find ways to encourage collaboration.

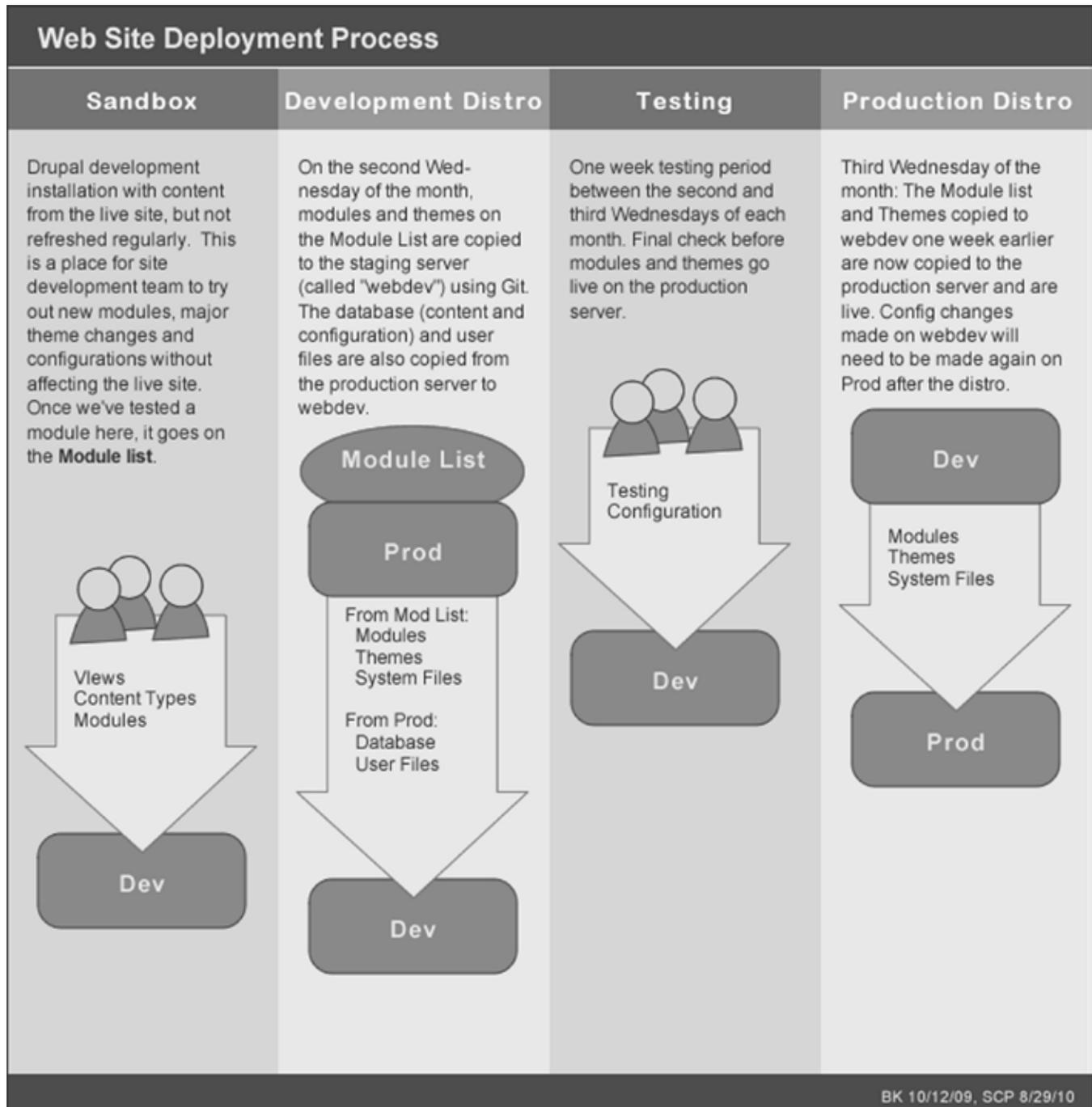
And the library is now in a much better position for future website design transitions, a process that will be much easier with so much less static content to migrate. For example, the look and feel of our entire website can be transformed by reconfiguring a few options within Drupal.

Ultimately, the transition of the library website to a Drupal environment was a very good thing, and we are glad we did it. It was difficult and messy at times, but our website is now more flexible, agile, adaptable, and better poised for change.

Epilogue

Since this article was submitted, the UC Santa Cruz University Library website has moved to an entirely new campus theme. We note that having a Drupal-based CMS greatly aided this transition process.

Appendix A. Website Deployment Process



BK 10/12/09, SCP 8/29/10

Created by Bryn Kanar and Sue Chesley Perry

Appendix B. Drupal Resources for Getting Started

- American Library Association. "Drupal4lib Interest Group (LITA - Library & Information Technology Association)." <http://connect.ala.org/node/71787> (accessed March 18, 2011).
- American Library Association. "Showcase: Database Pages & Research Guides Using Drupal." <http://connect.ala.org/node/98546> (accessed March 18, 2011).
- Austin, Andy, and Christopher Harris. "Drupal in Libraries." *Library Technology Reports* 44, no. 4 (2008).
- Byron, Angela, Addison Berry, Nathan Haug, Jeff Eaton, James Walker, and Jeff Robbins. *Using Drupal: Choosing and Configuring Modules to Build Dynamic Websites*. Sebastopol, CA: O'Reilly, 2008.
- Drupal. "Drupal.Org." <http://drupal.org/> (accessed March 18, 2011).
- drupal dojo. "Drupal Dojo." <http://drupaldojo.com/> (accessed March 18, 2011).
- Drupal Modules. "Search, Rate, and Review Drupal Modules." <http://drupalmodules.com/> (accessed March 18, 2011).
- "DRUPALCONSF San Francisco – April 19-21, 2010." <http://sf2010.drupal.org/conference/sessions> (accessed March 18, 2011).
- drupalib. "drupalib: A Place for Library Drupalers to Hang Out." <http://drupalib.interoperating.info/> (accessed March 18, 2011).
- GotDrupal.com. "Gotdrupal: Once You've Got It, You're Addicted!". <http://gotdrupal.com> (accessed March 18, 2011).
- Groups.Drupal. "Libraries." <http://groups.drupal.org/libraries> (accessed March 18, 2011).
- Groups.Drupal. "List of Libraries Using Drupal." <http://groups.drupal.org/libraries/libraries> (accessed March 18, 2011).
- "Is This Site Built with Drupal?". <http://www.isthissitebuiltwithdrupal.com/> (accessed March 18, 2011).
- Learn By The Drop. "Learn by the Drop: A Place to Learn Drupal." <http://learnbythedrop.com/> (accessed March 18, 2011).
- "Lullabot." <http://lullabot.com> (accessed March 18, 2011).
- Mastering Drupal. "Drupal Screencasts." <http://www.masteringdrupal.com/videos> (accessed March 18, 2011).
- Slideshare. "Drupal Resources for Libraries, Sarah Houghton-Jan." <http://www.slideshare.net/librarianinblack/drupal-resources-2982935> (accessed March 18, 2011).
- Slideshare. "Introduction to Drupal for Libraries, Laura Solomon." <http://www.slideshare.net/OPLIN/intro-to-drupal-for-libraries> (accessed March 18, 2011).
- SUNRAINPRODUCTIONS. "DrupalCampLA 2009 - Views Demystified." <http://www.sunrainproductions.com/drupalcampla/views-demystified> (accessed March 18, 2011).

Appendix C. Selected Drupal Modules Used on the UCSC Library Site

- **Administration Menu**—adds a top menu bar for authenticated users with common administration tasks
- **CCK**—allows you to add new content types, for example the online resources content type for A–Z list
- **CKEditor**—WYSIWYG editor
- **Google Analytics**—adds Google javascript tracking code to all of our site's pages
- **Google CSE**—allows us to use Google as the site search
- **IMCE**—image-uploading module, also allows you to create subdirectories within the image directory
- **Image Cache**—allows you to pre-set sizes for images
- **LDAP Integration**—links user authentication to the library's LDAP server
- **Mollum**—spam filter and Image CAPTCHA (part of Spam control)
- **Nice Menus**—allows drop-down/right/left expandable menus
- **Nodeblock**—allows you to specify a content type as being a block, which content creators to edit the block text and title without having to access the block administration page
- **Pathauto**—automatically generates path aliases for various kinds of content (nodes, categories, users)
- **Printer-friendly, e-mail and PDF versions**—allows you to configure any type of page to display links for print, e-mail, and PDF
- **Rules**—allows site administrators to define conditionally executed actions based on occurring events, we use it to send email when new content is created and to hide some content fields from selected user roles
- **Taxonomy**—enables us to assign subjects and other categories to content; the url paths and Views use taxonomy
- **Webform**—enables quick creation of forms and questionnaires