About Ohloh.net stats: "Since drupal.org moved from CVS to Git, this summary contributions project is no longer representative of contributed activity because each project has its own repository instead of one, large shared one. Ohloh added a Drupal contributions logo for maintainers to list their projects separately."

Similar

- Both are full-featured free and open source web applications with which you can realize a very diverse variety of projects.
- Both are written in PHP and use a database such as MySQL
- Both can run on standard, inexpensive hosting.
- Both use jQuery
- Both are community-run projects with a commercial ecosystem
- Both dogfood for the vast majority of project needs (Collaboration, Publishing, etc.)

Different

- Tiki uses Zend Framework and the Smarty Template engine while Drupal has its own code.
- Tiki has a predictable 6-month release schedule, with Long Term Support versions, while Drupal is released "when it's ready". Between Drupal 6.0 (Feb 2008) and Drupal 7.0 (Jan 2011) (nearly 3 years)
  - Contributors to Tiki can count on their code being part of an official stable release within a predictable and rapid time frame.
- "Any feature that we add to Drupal core needs to be good enough to last us at least 3 years in a relatively unchanged state"
  - In Tiki, we want things in core as soon as possible, because that is how they get better. Unstable features can be tagged as "experimental".
- Drupal is more of a Content Management Framework, whilst Tiki is more of an out-of-the box Content Management System with all the features built-in. Because of these design choices, Tiki has more features out-of-the-box. With Drupal, you can add tons of features via modules.
- As you can see in this case study, Tiki code is much simpler (more like standard PHP), vs Drupal's multiple abstractions.
- Tiki is more centralized. Tiki is the all-in-one model while Drupal (like Joomla!) is the small-core-and-add-what-you-need model. Each model has its pros & cons.
  - In Drupal, there can be more than one "contributed module" for the same feature so they have different names. Whereas in Tiki, there is only one and thus, the name is descriptive.
  - Tiki has all the features built-in (and you just activate/deactivate features), thus, every Tiki instance of a given version has the same code base. This makes it easier for a hosting company and for upgrades. In contrast, if you maintain dozens of Drupal sites, they will have different modules installed (and thus, code base) depending the use case.
  - When Tiki is upgraded, all the features are supported and the upgrade is smooth. In Drupal, some of your plugins/extensions may have become abandoned or be incompatible with the new version.
- In Drupal, there is a clear distinction between core and contributed modules developers. In Tiki, everyone works on the core.
- While Tiki is a very large and popular project (over 250 contributors to main code base), Drupal is **several times larger** when you count the contributed modules.
- The Wiki engine in Tiki is mature, very powerful and versatile and has features hardly any wikis have such as the cross lingual wiki engine, whereas, the Wiki modules (and again the problem of choosing which one) for Drupal are more modest and less actively developed.
- ~~Drupal has modules for features that Tiki doesn't have, ecommerce for example~~ Tiki5 now has a basic shopping cart.
- Drupal is GPL, Tiki is LGPL. The main difference between the GPL and the LGPL is that the latter can be linked to (in the case of a library, 'used by') a non-(L)GPLed program, which may be free software

or proprietary software.
- Drupal has a requirement that all code hosted on Drupal.org be GPL. Whereas Tiki will routinely re-use code of any license that is compatible with the license used by Tiki.

Complexity

- The Tiki code base is much much simpler
  - Since the beginning, Tiki has had a focus on keeping the code as simple as possible to make it easy to customize and easy for new developers to participate.
  - A case study of Drupal, Tiki Wiki and XOOPS demonstrates how much complex the Drupal architecture is
  - http://benbuckman.net/drupal-excessive-complexity
  - https://archive.davidherron.com/davidherron.com/content/drupals-complexity-turning-people-hurm.html
  - See more how Tiki is Coping with Complexity.

Differences about upgradability

- Tiki is much easier to upgrade.
  - *.tiki.org sites are always running the latest stable or LTS version as this helps improve quality and performance. In fact, *.tiki.org sites are upgraded to new versions (thus twice a year) in the alpha and beta stage *before* the new release becomes official. By contrast, over a year and a half after the release of Drupal 7, drupal.org is still running Drupal 6: "To help pull it all together Angie 'webchick' Byron compiled a team of 23 awesome developers from across the community.". This is yet more proof on how much easier an all-in-one app (like Tiki) is to upgrade. Tiki even has Pre-Dogfood servers so the community can check daily on upgrade success!
  - Both Tiki & Drupal cover many many use cases. To make it easier for users to get started, "out of the box" use cases (ex.: Intranet, Community site, etc.) are covered by "distributions" in Drupal, while they are "profiles" in Tiki.
    - Drupal distributions are packaged as a .zip and include Drupal, the modules, themes, etc. They are for installation only and not designed to be combined.
    - Tiki profiles are recipes that can be applied on an existing Tiki installed (they can be combined)
      - Because Tiki profiles do not contain any custom PHP code (that part is in Tiki), but merely configuration and content, sites deployed via profiles are easy to upgrade later on. (The CartoGraf profile does require you to download a theme though)
        - In contrast, because Drupal distributions contain various modules, they can be difficult to upgrade. For example, one of the most popular ones (Open Atrium) took [https://www.drupal.org/node/2118835|2 years and 9 months to release a version for Drupal 7.
          - febbraro (CTO of the company maintaining Open Atrium) explains: "I'm not sure how many sites or even modules you have upgraded to D7 from D6, but I can tell you it is one hell of a heavy lift and after doing it a few times we are definitely looking first before we leap. To think about doing this for a product customized as much as Atrium is a huge and costly undertaking that we at Phase2 cannot carry the burden alone (as other seem to suggest we should). What we can promise is to be more open about the direction we think the product should take, and what we can contribute to the effort."
- For us, by far the biggest problem with Drupal is upgrade pain — the single biggest reason we're losing customers these days is sticker shock at what it takes to upgrade from D5 or D6 to D7.

CMS Match

http://www.cmsmatch.com/compare/content-management-systems/11+1073

CMS Matrix

Compare them at http://cmsmatrix.org

Openhub comparison

https://www.openhub.net/p/_compare?project_0=Drupal+%28core%29&project_1=Tiki+Wiki+CMS+Groupware&submit_1=Go

Contributors (All Time) click "View as graph"
Commits (All Time) click "View as graph"

Drupal has a very large number of non-core developers and the stats for them do not appear here.

The charts above don't take into account the huge activity for contributed modules. As it's almost always the case with extension-based systems, there are vast numbers of contributed modules (which shows activity and diversity), but they often overlap if not outright duplicate each other or can be abandoned. See Tiki all-in-one model for more information. Also, in Drupal, contributions are suggested by a larger number of people but the commits are done by a smaller number of people. Thus, the chart above is skewed.

Related links

- DrupalCon DC 2009 - Why I Hate Drupal (James Walker). Note: Drupal is a hugely successful project. He describes the challenges of the Drupal model. Each model has strengths & challenges.
- Product, Framework, or Platform? What They Mean, And Why You Should Care -> This talk highlights the challenges of attempting to be many things at once.
- Seducido por Tiki Wiki: estoy siendo infiel a Drupal - Great article about how long-time Drupal supporter has been "seduced" by Tiki. (in Spanish)
- How CMS architecture affects dev communities - A case study of WordPress, Drupal, Tiki Wiki and XOOPS
- http://erickennedy.org/Drupal-7-Reasons-to-Switch
- Interesting Features from other Web apps

alias

- TikiWiki vs Drupal
- Drupal