

Against the Grain

Volume 22 | Issue 1

Article 14

February 2010

Under the Hood -- Feedback Loop: An Interview with Jeff Dietrich, Cou tts Information Services

Xan Arch

Stanford University Libraries, archx@reed.edu

Follow this and additional works at: <https://docs.lib.purdue.edu/atg>

 Part of the [Library and Information Science Commons](#)

Recommended Citation

Arch, Xan (2010) "Under the Hood -- Feedback Loop: An Interview with Jeff Dietrich, Cou tts Information Services," *Against the Grain*: Vol. 22: Iss. 1, Article 14.

DOI: <https://doi.org/10.7771/2380-176X.5844>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

Under the Hood — Feedback Loop

An Interview with Jeff Dietrich, Coutts Information Services

Column Editor: **Xan Arch** (Electronic Resources and Technology Librarian, Stanford University Libraries, Stanford, CA 94305-6004; Phone: 650-725-1122; Fax: 650-723-4775) <xanadu@stanford.edu>

The point? Just this: try as these stinkers might, a Day in the Sun is the best many of them can hope for. They can invent and market the stuff — they can seize ascendancy — but at the end of the day they either go the way of the dinosaurs, killed off because they could no longer handle the world, or the way of the Dodo, killed off by some predator giving no thought whatsoever to the consequences.

The ideas are ours. Somebody might own the printing press. We may have to enter into some contractual arrangement for our ideas to be permitted access to their reproductive machinery. Or, we can lease access to the Internet from a service provider and self-publish. But be warned: have a good enough thought, and you’re in the cross hairs. Someone may decide it’s far less trouble to steal your thought than to make up one of their own. Or, they may condescend to permit you access to the means of distribution to spread your thoughts and ideas far and wide — as long as they get a piece of the action.

But let’s never confuse the unveiling of a carefully constrained, carefully controlled, ruthlessly protected market introduction with the act of creativity.

What can we do?

Let’s fill the world with simple, inexpensive, light-weight machines running a decent open source operating system — netbooks running Ubuntu, for example — and make THEM the target for a million streams of wonderful, creative content, Project Gutenberg files, MIDI files, great recordings, timeless orations, and yes, even proprietary content you rightfully need to pay for to get at.

Just, please, oh great marketers of the world, please.

Stop telling us that because you’ve got a roller coaster, you’ve invented the delightful interplay between mass and gravity.

Stop trying to persuade us that knowing how to make a plastic bread bag means you’ve discovered the miracle of the leavening process.

Stop trying to palm off a flask as if you’ve invented the realm of spirits. 🍷

One of the threads of my last two columns has been feedback, primarily from librarians to vendors of library products. I’ve spoken about how this feedback needs to be structured in a way so it is clear and usable for the vendor. But is this all that is needed? What does feedback look like from the vendor’s perspective?

For this angle, I’ve asked a guest to join me. **Jeff Dietrich** is a Senior Software Engineer at **Coutts Information Services** and someone with a fluent understanding of the interaction between librarians and the people who build library software. He has managed the **OASIS** engineering team and now works on a broader range of software tools for **Coutts**.

XA: *What kind of feedback does your team receive?*

JD: It really runs the gamut: detailed feature requests, bug reports, would-be-nice-if suggestions, and the occasional furious denunciation. We definitely don’t get as much user input as we’d like to see. What we do see is always welcome and useful, if often incomplete. Users sometimes assume developers are more all-seeing and all-knowing with respect to application activity than they actually are, to the point of not mentioning where in the application a problem occurred, which list or ISBN was involved, what the error message said, and so on.

XA: *Why does this feedback matter?*

JD: Because user input is the single biggest driver of development decisions, as it should be. I am sure that nearly every **OASIS** user who has spent significant time with it has had creative thoughts and ideas about it. They are keenly aware of those little things that would save time and make things easier, and they no doubt hold opinions about how the way we implemented Feature X is boneheaded, etc. But only a minority of those users take the time to reach out, to engage and collaborate with us on improvements. One of the key ways in which this sets all of us back, is that we as developers often see a clear need for the same improvements and features, since we are heavy **OASIS** users too. But a developer with a dream does not a mandate make. If users thinking along the same lines were speaking their minds, we’d have stronger cases to apply resources to the things that matter to them.

XA: *How can libraries structure their input to be more usable?*

JD: Detail and context. In-depth user stories and perspectives from the trenches. We need these to understand better how librarians use the application day-to-day, and where the workflow bottlenecks are. It’s a useful starting point that a user wants to be able to do X, but the really useful information often hides in *why* they want to do it, and in their creative speculation about what the outcome might look and work and feel like. We can often work backwards from the underlying end goal, and find better solutions than are apparent if we simply take “do X” at face value.

XA: *How can libraries make sure their input is given attention and made a high priority?*

JD: Feature requests are not easy to generalize about as that goes — they all get prompt attention and generate internal discussion, in any case. When it comes to bugs, step-by-step reproduction cases are the most crucial. If we can reproduce a problem, it can typically be zeroed in on, fixed, tested, and included in a release relatively quickly. Without reproduction steps or enough detail to quickly establish them, precious development time gets burned trying to re-assemble the circumstances of the bug through log analysis, broad review of potentially relevant code, etc. Reports that will cost all of this extra effort to unravel are typically de-prioritized if they are not deemed critical. As more users report the problem, more internal staff become involved, more details become apparent, and the priority ticks upwards. But having better information up front cuts right through all of that. The right degree of detail provided by users can mean the difference between a bug being fixed next week, or three months from now.

XA: From **Jeff**’s perspective, the fundamental needs for feedback are not just clarity and structure, as I’ve discussed previously. The steps needed to reproduce a bug and the details of why a feature would help a library’s work are also elements that make for a valuable contribution to a vendor’s product. These contributions benefit the vendor, of course, but also the library who requests the change and the wider community that will find the change useful. While **Jeff** comes from work on **OASIS**, an on-

line ordering system, this holds true for the other systems used in libraries, from the ILS to link resolvers, even to social media tools. If we want these products to work for us, we have to start by asking for change. Thanks **Jeff**, for your feedback to librarians! 🍷

