

---

# Release 2.0<sup>1</sup>

Issue 2.0.1, February 2007  
<http://r2.oreilly.com>



"An astute observer can note the faint signals of the coming future, extrapolate from them, and begin to divine the business implications."

Tim O'Reilly, *From 1.0 to 2.0*, page 1

## Release 2.0

Premier Issue February 2007

Published six times a year by  
O'Reilly Media, Inc.,  
1005 Gravenstein Highway North,  
Sebastopol, CA 95472  
<http://r2.oreilly.com>

**This newsletter covers the world  
of information technology and  
the Internet—and the business  
and societal issues they raise.**

executive editor  
Tim O'Reilly  
[tim@oreilly.com](mailto:tim@oreilly.com)

editor  
Jimmy Guterman  
[jimmy@guterman.com](mailto:jimmy@guterman.com)

publisher  
Sara Winge  
[sara@oreilly.com](mailto:sara@oreilly.com)

art director  
Mark Paglietti  
[markp@oreilly.com](mailto:markp@oreilly.com)

copy editor  
Laurel Ruma  
[laurel@oreilly.com](mailto:laurel@oreilly.com)

contributing writers  
Brady Forrest  
Nathan Torkington

© 2007, O'Reilly Media, Inc.  
All rights reserved. No material  
in this publication may be  
reproduced without prior  
written permission; however, we  
gladly arrange for reprints, bulk  
orders, or site licenses. Individual  
subscriptions cost \$495 per year.

subscription information  
Release 2.0  
PO Box 17046  
North Hollywood, CA 91615-9588  
<http://r2service.oreilly.com>

customer service  
1.800.889.8969  
1.707.827.7019

# Contents

- 01:** From 1.0 to 2.0  
Keep an eye on the radar. The future is already here, if you look in the right places.  
Also: How the Cycles Repeat  
**By Tim O'Reilly**
- 04:** Under new management  
Your editor discusses what's changed, what hasn't, and how the title for the new  
newsletter may seem trendy but it goes back all the way to 1993.  
**By Jimmy Guterman**
- 06:** Web 2.0...  
It's the meme of the moment. It's the subject of everything from excitement to  
parody. In this issue, we take some aspects of the business revolution and look at  
them from some unusual angles.
- 07:** A Simple Story  
What if Web 2.0 stopped being a buzzterm and started making sense?  
What if we started considering Web 2.0 as something perfectly normal  
and reasonable?  
**By Nathan Torkington**
- 08:** People or Computers?  
There's more than one way to do things. And Web 2.0 provides a  
fascinating framework for comparing machine-generated data with what  
humans can generate—sometimes without meaning to.  
**By Brady Forrest**
- 12:** More. Better. NOW!  
Continuous iterative development isn't just for technologists. It's for  
businesses, too.  
**By Jimmy Guterman**
- 16:** The Number: One Person Per Blog
- 18:** The Canon: *Designing Interactions*
- 19:** Coming soon
- 20:** Calendar

---

# From 1.0 to 2.0



Tim O'Reilly is the founder and CEO of O'Reilly Media, Inc., and an activist for open standards. Tim's blog, at <http://radar.oreilly.com>, "watches the alpha geeks" to determine emerging technology trends, and serves as a platform for advocacy about issues of importance to the technical community.

We're pleased and humbled to take the torch from Esther Dyson on one of the computer industry's lasting institutions, the *Release 1.0* newsletter. Because we want to mark the fact that there is a new team at the helm, as well as our belief that the center of gravity in the industry has moved from the personal computer to the internet platform that has come to be known, for better or worse, as Web 2.0, we've decided to increment the title, and call this next stage in the life of the newsletter *Release 2.0*. O'Reilly originated the term Web 2.0, but this doesn't mean that we're going to focus only on the business implications of Web 2.0 and related technologies. Far from it.

The core of our business at O'Reilly is the notion that world-changing technologies often start not with entrepreneurs but with enthusiasts. Whether it's the first Apple I prototype, crafted in a woodshop by Steve Wozniak; the first skateboards, built by wanna-be surfers for riding the cement waves of unfinished swimming pools in Southern California; or the first websites, built to share scientific and cultural documents, new industries are often launched inadvertently by people who try new things because they love them, not because they think they are going to get rich.

This is the "alpha" stage of the industry. Here we fondly refer to these enthusiasts building prototypes of the future as "alpha geeks," not only because of the early stage nature of what they do but also because, like the alpha male in a wolf pack, they are often the strongest. The alpha geeks stand out in any gathering of technologists as those who are the masters of what they do, able to bend technology to their will and vision, rather than accepting it ready made.

Watching these alpha geeks is the core of what, on our blog (<http://radar.oreilly.com>), we call the "O'Reilly Radar." The alpha geeks demonstrate, again and again, the truth of William Gibson's dictum that "The future is here. It just isn't evenly distributed yet." When Tim Berners-Lee wrote the first web browser and web server, he had no idea of what he was setting in motion. ->



“We hope to give you insight into this entire cycle: What are the alpha geeks doing today that will be crucial to your business two years from now?”

-> He just wanted to help the physicists at CERN share their work. When Linus Torvalds decided to write a new operating system kernel “just for fun,” he had no idea how Linux would shake up the power structures of the entire computing world. When Shawn Fanning wrote a program to help his friends share their music, he had no idea how the P2P revolution he ignited with Napster would transform not just music, but also television and publishing.

But an astute observer could note the faint signals of the coming future, extrapolate from them, and begin to divine the business implications of what would one day be a new world accessible to all.

### Alpha geeks enter beta...and beyond

And so, the beta phase begins. Alpha geeks become entrepreneurs. The venture capitalists arrive. No one knows yet what will work, but the scent of opportunity is powerful.

After a few years, the rules of business are clear. Early entrepreneurs play leapfrog, innovators hit on winning formulas, and gradually, a new industry emerges. You might call this the “1.0” phase.

But all is not well. If a technology is truly transformative, it attracts too much capital, too many me-too entrepreneurs, and the result is inevitable: bubble, followed by crash. Argentinean economist Carlota Perez, now at Cambridge University, has traced this pattern through history, and has seen it repeated again and again: with canals, railroads, steel, cars, and computers. She has found that the bubble is an essential stage in the formation of a new industry, and is typically followed by a “long boom” in which the new infrastructure built during the bubble is put to real use by a next generation of entrepreneurs. This is the era, broader than Web 2.0 but including it, that we’ll chronicle in *Release 2.0*.

In this newsletter, we hope to give you insight into this entire cycle: What are the alpha geeks doing today that will be crucial to your business two or more years from now? Who has begun to understand the new rules of business, the unexpected transformations of leverage that will create new winners and losers? What are those rules? And whether you’re an entrepreneur, an investor, or a corporate technology strategist, how can you apply them to create value for your business?

If we do our job, you’ll be surprised, you’ll be inspired, you’ll be prepared for the future that is already here, if you look in the right places. ■

## How the Cycles Repeat

With the dot-com bust so recent, it's easy to forget that the PC business had its bubble and its shakeout, and its roster of high-flying companies now long forgotten. But it's also easy to see how, over a period of two decades, the PC moved from enthusiast's toy to the center of the computer industry. Looking back at the history of the PC, you can discern a similar trend to what's happening now in the internet industry.

A new stage in an industry is characterized not just by new technologies and new companies, but also by new levers of competitive advantage. Steeped in a history in which value was rooted in computer hardware, IBM and its peers had no insight into how PCs built from standard off-the-shelf parts would lead to the commoditization of hardware, and (via what Clayton Christensen calls "the Law of Conservation of Attractive Profits") to a massive shift in business leverage from hardware to software. And so it signed away its future to a small company called Microsoft. Fifteen years later, history repeated itself. Convinced of the value of proprietary software, Microsoft had little insight into the ways open source software and standardized internet protocols would lead to the commoditization of software and the transfer of power in the computer industry to new information businesses harnessing advertising-based business models. Now, like IBM in the late 1980s, Microsoft is rushing to catch up.

At the mid-point in an industry, when competition appears to be fiercest, the company that best understands the new rules of business may have already passed a point at which its eventual ascendance is virtually guaranteed. Old-timers might remember that it was also the late 1980s when Microsoft and IBM (which was in catch-up mode with OS/2) appeared to be locked in a struggle for dominance, but the battle was over. Microsoft had offered application developers an attractive, if Faustian, bargain: forget about writing all your own drivers, just use our APIs. With that move, Redmond's platform dominance was assured, and over time, it was able to use that platform to drive out competitors and raise the barriers to market entry. Entrepreneurs had to go elsewhere or build a business plan with "acquisition by Microsoft" as the sole exit plan—just as countless startups today dream of being bought out by Yahoo! or Google.

So the story begins again, with alpha geeks who build for love, not money. While investors and existing players were focused on the PC application and platform wars, the next generation of disruptive technology was being built in a lab in Switzerland and in a dorm room in Helsinki. Just when it looked like it was game over, with a dominant monopoly sucking all the opportunity out of the ecosystem, a next generation of entrepreneurs was readying the new business models that would reset the game. —**T.O.**





Jimmy Guterman is the editor of *Release 2.0*. Previously, he was editor at *Forrester*, *Gaming Industry News*, *Media Grok*, *CD Review*, and elsewhere. He can be reached at [jimmy@guterman.com](mailto:jimmy@guterman.com).

“We will offer early intelligence on ideas and companies before they explode into the public consciousness—and a context for understanding what they are going to change and how they are going to do it.”

# Under new management

*What's changed, what's not.*

**by Jimmy Guterman**  
**February 12, 2007**

I'm your new editor, but I don't feel new. I have been a regular reader of *Release 1.0* since 1993 and a paid subscriber since 1995. I've waited for the return of the newsletter as much as you have, so I welcome you back with a combination of relief and delight. *Release 1.0* was a terrific early-warning guide for new trends and technologies, even if it was unclear to some (most?) of us at first how important various developments and discoveries would be. I know I've missed the newsletter. I don't feel as smart without it.

I know you've missed it, too, because you told me. I've spoken to more than 100 subscribers in recent weeks. (If we haven't corresponded yet and you'd like to, I'm at [jimmy@guterman.com](mailto:jimmy@guterman.com).) I've learned how devoted you remain to a publication that hasn't appeared for the better part of a year. I've also been reminded that you're an opinionated bunch. Some of you had specific requests (yes, we've brought back the calendar and will expand it online); others wanted me to stop talking to readers and start producing the newsletter. There were several points that many, if not most, of you brought up. I'll address some of those here.

Speed, please.

No one I talked to was happy when *Release 1.0* bumped down from almost-monthly to quarterly. We're increasing the frequency of publication to six times this year and we'll return to almost monthly delivery next year, with a little summer vacation built in for all of us.

Not too much speed, thank you.

We work in an industry where ideas and businesses go from rumor to much-ballyhooed launch to much-copied cliché in record time. There are some fascinating horse races going on right now, but *Release 2.0* is not in the business of following horse races in progress. We hope to offer early intelligence on ideas and companies before they explode into the public consciousness—and a context for understanding what they are going to change and how they are going to do it. In particular, we'll be investigating how companies are using data in new ways.

Even when you go back to monthly, that's a long time between issues.

One subscriber told me, "I don't read newspapers or magazines anymore. I don't even visit websites anymore. I read RSS feeds." Like many of us, he wants to know what he needs to know right now, and then he moves on, skimming all

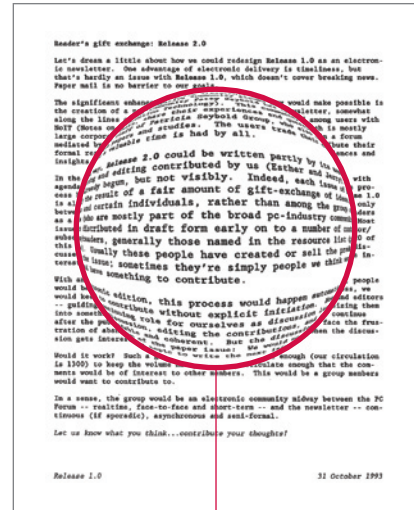
else. A newsletter that visits you only 11 times a year isn't holding up its end of an ongoing conversation. As this year rolls on, we will increase dramatically the online presence of *Release 2.0*.

About that name...

Yes, O'Reilly Media was where the term "Web 2.0" was born. It was intended, in part, as an umbrella under which we could understand several exciting and related developments. And, yes, the term is being misused everywhere from newspapers to business plans, usually by people trying to sell something, arguing that this boom is more rational than the last one (that's an open question). But Web 2.0, we maintain, is about continuity as well as disruption. As David Weinberger (a *Release 1.0* contributor and author of the forthcoming *Everything Is Miscellaneous*) notes, "Web 2.0" sounds like a second chance. But the story is essentially misleading, especially if applied outside of the economics of the web. While businesses may have gone through a boom and bust cycle, the rest of the web has continued doing what it does: connecting people, making what people create accessible to everyone, and innovating, innovating, innovating."

*Release 1.0* was full of great writers and thinkers, not just regulars Esther, Jerry Michalski, and Kevin Werbach, but also John Seely Brown, Dan Gillmor, our own Tim O'Reilly, Clay Shirky, and many others. I write most of this issue, but that won't be the case every issue. *Release 2.0* is a team effort. Anyone familiar with O'Reilly's Radar blog (<http://radar.oreilly.com>) knows that O'Reilly Media is filled with smart, opinionated people who hear about Next Big Things before most of us. In this issue we hear from Radar members Brady Forrest, Tim O'Reilly, and Nathan Torkington. In the months to come, expect to benefit from the insights of the entire team.

At its best, *Release 1.0* felt like a wise, friendly voice whispering in your ear about what was coming around the bend. We too are hungry to learn what's coming next and share it with you. As much as we are looking forward, we do promise continuity. In the first issue of *Release 1.0* that I read, way back in October 1993, Esther Dyson wrote, "Let's dream how we could redesign *Release 1.0*." Her suggested name for such a publication: *Release 2.0*. Let's take her up on it. ■



#### Reader's gift exchange: *Release 2.0*

Let's dream a little about how we could redesign *Release 1.0* as an electronic newsletter. One advantage of electronic delivery is timeliness, but that's hardly an issue with *Release 1.0*, which doesn't cover breaking news. Paper mail is no barrier to our goals.

The significant enhancement which electronic delivery would make possible is the creation of a participative community around the newsletter.

With an electronic edition, this process would happen automatically; people would be able to contribute without explicit initiation. Nonetheless, we would keep a continuing role for ourselves as discussion leaders and editors—guiding the discussion, editing the contributions, and synthesizing them into something readable and coherent. But the discussion would continue after the publication of the paper issue: We would no longer face the frustration of abandoning a topic to write the next issue just when the discussion gets interesting!

[Excerpts from *Release 1.0*, October 1993]

---

# Web 2.0...

*As its technologies and practices turn to the mainstream rapidly, people are either attempting to define Web 2.0 or complaining that it's all marketing-speak. Everyone from Jeff Bezos to the blogger next door has made an attempt. Rather than try to define it once again, in this issue we examine Web 2.0 from some unusual angles. We'll start this three-part section with a reality check from O'Reilly's Nathan Torkington.*



---

# A Simple Story

*What if Web 2.0 stopped being a buzzterm and started making sense?*

**by Nathan Torkington**

We need to think about Web 2.0 so that it seems to be what it is: perfectly normal and reasonable, not a buzzword got mad. Don't think of Web 2.0 as something earth-shakingly new, but as a name for something that is already going on. The Web 2.0 story is a simple story, built on half a dozen key trends, each of which leads to the next:

1. Applications have moved to the web. This frees companies from having to ship executables around the world on discs.
2. These web applications, from Salesforce to Gmail, have evolved richer interfaces to resemble desktop applications.
3. These applications are built with the intention of letting individuals and companies access and manipulate their data without the data being forced onto the same hard drive as the application.
4. That data accessible via the web isn't merely unlocked from a particular hard drive; it is now available from any device with a connection to the net.
5. Because the applications are served centrally, developers can continuously improve their applications and enjoy a competitive advantage from doing so.
6. Because the data used by these applications is stored centrally, rather than on a particular device, developers can use aggregated data to make applications better for all comers. Web 2.0 moves the focus from someone else's application to your data.

Those trends are all powerful, fueling the current boom or bubble or whatever you want to call it. Those trends are all simple, explainable in short declarative sentences. Yet there is a desire by many purveyors of fear, uncertainty, and doubt to make those trends seem much more complicated than they are. Don't let them get away with it.

Behind the technology are new opportunities for business. Every step in this move to web applications responds to needs from customers, and where customers have needs there are businesses to be built. It's not just programmers who should be thinking about Application Programming Interfaces (APIs). The group at Amazon.com responsible for APIs created \$200 million in revenue last year. A full 60 percent of eBay's listings, the primary source of the company's revenue, come from sellers using its API.

Web 2.0 is not a cloud of random concepts. It simply describes the natural consequences of the movement of applications from desktop to web. Why should that be so hard? ■



Nathan Torkington's crimes in the computing community include: coauthor of the *Perl Cookbook*, editor for O'Reilly, content coordinator for the Open Source Convention and Perl Conference, and project manager for perl6.

For those who do want a definition, try this one:  
*Web 2.0 is the business revolution in the computer industry caused by the move to the internet as platform, and an attempt to understand the rules for success on that new platform.*



Brady Forrest is co-chair for O'Reilly's Web 2.0 Expo, Where 2.0, and Emerging Telephony conferences. He previously worked at Microsoft on Live Search (he came to Microsoft when it acquired MongoMusic). Brady lives in Seattle, where he builds cars for Burning Man and runs Ignite! (<http://igniteseattle.com>).

---

# People or Computers?

*Data is at the core of Web 2.0. Now let's argue about the best way to capture it.*

**by Brady Forrest**

Anyone familiar with the popular web programming language Perl has heard the motto "There's more than one way to do it." Problem solvers know that there are different ways to reach the same goal. It's a motto that fits well with the open source aesthetic, but sometimes picking one approach means choosing sides. Choosing sides means battles. And battles often have a winner and a loser.

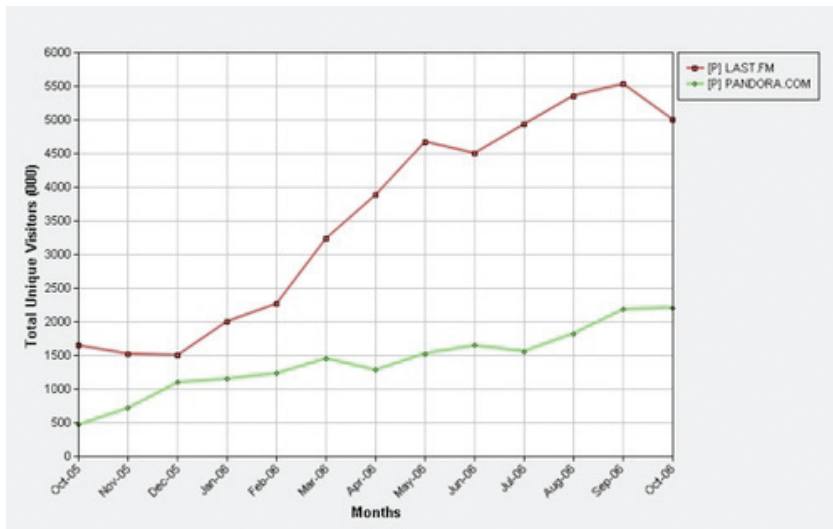
Manipulating data in new and interesting ways is a keystone of Web 2.0. The two primary approaches for developing data is either through the work of people (the grossly named "user generated content," often abbreviated as UGC) or algorithms. Digg's top headlines are a result of what its members have voted on; Google News's headlines are the work of decidedly nonhuman algorithms.

Yet the worlds of UGC and algorithms are not so easily separable. Digg, the human-generated news-sharing service, got its biggest boost early in its life because it was recognized by Google's algorithms. When Paris Hilton's mobile phone was hacked, Digg contributors were among the first report on it. The Digg article quickly rose to the site's front page, where Google and Yahoo! indexed it. The next day, as the news became more widely reported, thousands turned to search engines for "paris hilton cellphone hack." The top result was Digg and the site had its first mainstream hit.

## Last.fm versus Pandora

Last.fm relies on its members to populate its site, build the relationships between songs and performers, and determine what plays on personalized "radio" stations. In contrast, Pandora uses data produced by musicians and other experts and a set of algorithms originally designated "The Music Genome Project" to create custom stations for its members. There's more than one way to do it: two different approaches lead to similar services.

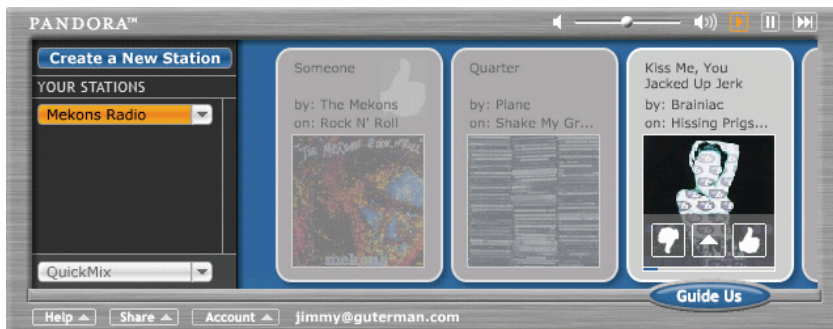
On Pandora's home page, you'll find a music player. Type in the name of a band or song (Portishead, for example), press play, and you will hear similar music (trip-hop, in this case). How does it do it? Musicians analyze a song and describe it using a set of agreed upon attributes. The Music Genome Project has developed a list of more than 100 attributes that can be used to define a song. Songs get rated and their individual ratings are rolled up to provide an artist score. Using this data and its own algorithms, the service is able to group similar-sounding songs together. →



According to Comscore, Last.fm spent much of last year pulling away from Pandora in terms of members. We first saw this comparison on Fred Wilson's "A VC" blog, at [http://avc.blogs.com/a\\_vc/2006/11/pandora\\_vs\\_last.html](http://avc.blogs.com/a_vc/2006/11/pandora_vs_last.html). To explain the difference, Wilson wrote, "Last.fm started off as a social network and evolved into a streaming music service. Pandora took the opposite approach."



Play a song by a band in Last.fm and you'll get a tag cloud filled with names of other bands you might like. How does Last.fm know? Because other members who liked the band you're listening to liked these bands, too.



Create a radio station inside Pandora around a particular band and you'll be greeted by a suggested playlist of other bands's songs. Those other songs are determined by how similar their algorithmic fingerprints are to the song you're playing.

### When you're on the Internet, no one should know you're a Crazy Frog fan

Ever show up for a first date with someone you met online and only to discover that the self-described 19-year-old surfer god was in his late 40s, bald, and paunchy? Online social networking is still in its infancy and the desire to appear better than we really are is as true in music services as it is in dating services. "People have guilty pleasures," says Last.fm's Martin Stiksel. "So they have to have some music in their profile that they're not keen on showing to everybody. They'll say, 'That wasn't me. Yesterday my little sister was on my computer.' Yeah, sure. Members have asked us to remove tracks, so we offered a feature to edit the most recent 200 tracks in a profile." So that little Barry Manilow addiction can remain a secret.

→ Last.fm's data comes from its members' listening habits. It gathers this data by "scrobbling," or tracking what individuals are listening to. Using "scrobbles" that plug in to various music players, Last.fm learns what's in your library, how often you listen to each song, and how you rate each song. Over time, Last.fm considers your ratings and modifies its suggestions based on what you've told it.

By contributing your ratings and other data that identifies you, you are building a profile. You can find out what people you know are listening to, which songs you have in common with people you don't know, and who's online now. The amount of information in this social network can be enormous. Martin Siskel of Last.fm says that the service takes in 500 million discrete pieces of information from its customers per month. Most of it is captured by the software, not explicitly asked for. Within clear limits, it tracks one particular activity—listening to music—and pulls in all sorts of data that may be idiosyncratic at the individual level but fascinating in the aggregate. This amount of music metadata also opens up a secondary business model for Last.fm—music data services. Social networks can be monetized, but perhaps those data services could develop into Last.fm's primary business, just as it did for CDDB when it became the for-profit Gracenote. With the personal data, Last.fm could also deliver ads targeted to members—not just the content on the page. They could be ads that customers asked for without even knowing it.

### Which way to success?

What about a middle path? Is it always a choice between experts and community? One way to get humans to perform the work of algorithms (or provide what Amazon calls "artificial artificial intelligence") is Amazon's Mechanical Turk, a marketplace for getting people to perform small tasks for miniscule amounts of money. There is a community doing the work, but the data production is planned centrally.

"To develop a compelling service, you often have to implement a combination of things," says Tim Westergren, founder and chief strategy officer of Pandora. "There are different approaches to making connections between listeners and each has merit. We always believed a combination of two would be necessary. One approach alone won't carry the day over the long run. A system using one approach won't be as good as system that does it all."

However Pandora develops, Westergren insists that there's no "religious" approach toward algorithms or anything else. "We're going to add anything we can to Pandora to make it better. We've started with collaborative filtering and user inputs and we'll do it more. Anyone who spends time slamming another technology should look in the mirror."

---

And let's not forget Google, the canonical example of an algorithm-based system. Think about it: What is PageRank, the crown jewel of its system, made up of if not content generated by individuals? It is algorithm-based but far away from algorithm-only. Perhaps Google's PageRank works so well because it is based, at its core, on human behavior.

Algorithms can be beautiful, powerful things. But in the Web 2.0 space, algorithm-based systems can get business only so far and may, in fact, be losing their sway. There may be more than one way to do it, but content generated by large numbers of people is often the most reliable and the most successful. (There are notable exceptions, of course. Non-algorithmic search systems, on the whole, have not taken off.) As James Surowiecki and others have argued, the crowd, when properly mined, turns out to be smarter than a small group of vetted experts. (As the famous album cover read: *50 Million Elvis Fans Can't Be Wrong*.) That's why the expert-driven Google Answers is gone, overtaken by the crowd-driven Yahoo! Answers. Last.fm could add algorithms to better process its data, but due to the law of increasing returns Pandora cannot catch up to the volume of data Last.fm can ingest. If the Comscore numbers are any indication (see chart on page 9), Last.fm is far more popular than Pandora. That feeds the service: The more people who use Last.fm, the more data it has. The more data it has at its disposal, the smarter the database becomes, the more useful the program is, and the more people will want to use it.

So much of the data captured by Last.fm is passively passed on by listeners, not explicitly captured and sent by people. Just as people using the original Napster had to contribute a list of every file in their music directory by default, users of Last.fm (or similar services such as Mog and iLike) send all sorts of useful information by default. They are building a database that has helped Last.fm make a deal with at least one of the major recording labels—and they're not even doing it on purpose. The most human of all activities—the behaviors we engage in when we're not even trying—may be the most powerful of all fuels for Web 2.0 products and services. ■■

---

### **Lost in space with Google Earth and Microsoft's Virtual Earth**

Both Google and Microsoft offer geobrowsers that let you fly around virtual worlds viewing satellite imagery. (It's like Second Life—with real maps!) Both are incrementally populating their respective worlds with 3D models, but the two have taken different paths in that quest.

Google offers SketchUp, a 3D modeling tool it acquired from atLast Software last year, and a website to show off and share its work. Some models have been vetted and are now viewable in Google Earth (itself developed at another company, Keyhole).

Microsoft is also using acquisitions to get its models. In 2005 it purchased Vexcel and GeoTango. Using Vexcel's image-capture technology and GeoTango's 3D visualization tool, GlobeViewer, Redmond is beginning to build out its world.

Geobrowsing is in some ways tougher than music. Incorporating amateur data is more difficult and the cost of acquiring expert data is much higher. In Last.fm, if multiple people report that they like both the Beatles and the Rolling Stones, it reinforces the connection. But if a geobrowser has access to 10 models of the Empire State Building, that's nine too many. Pandora merely needed to pay some musicians to analyze music, not have planes fly over 15 cities.

Microsoft's approach helps it better target the population centers, but it will not be cost-efficient for Virtual Earth to cover many rural parts of the country. Google will incur no additional cost to get the entire country done, but it won't know when (or if) it will get done, because, unlike Microsoft, it does not control its contributors.

---

# More. Better. NOW!

*Continuous iterative development is one of the mantras of Web 2.0, but its influence goes way beyond.*

**by Jimmy Guterman**

“People expect constant improvement now.”

— Marc Lucofsky, Google

Progress is about learning from those who came before you—and coming up with something new despite your debt to them. Elsewhere in this issue, we look at Bill Moggridge’s brilliant *Designing Interactions*, in which new generations of giants build on the achievements of giants from previous generations. Progress can mean tearing down, too, and many innovators do so by denying their forebears. The renegade film directors of the 1970s had harsh words for the studio system they displaced, the Sex Pistols denied everything (despite delivering a pretty good Monkees cover), and during both the ‘90s boom and the current one, there has been no more damning epithet than saying someone or something was from an earlier time and “doesn’t get it.” We’ve all heard someone dismiss a product or service as “Web 1.0.”

Imagine a Web 2.0 application (the kind that TechCrunch leaks, DEMO debuts, GigaOm celebrates, and Valleywag dismisses). Whenever something wrong happens in the application, you can send a message to the developer, along with nonprivate information the application has saved about what went wrong. Those reports enter a database, that database senses patterns, and if there is a problem with the application that’s due to more than your idiosyncratic setup, a bugfix is sent right back to you. Whatever company came up with something like that would have to “get it” and understand how Web 2.0 works, right?

Well, the company behind such a Web 2.0 application is Microsoft and the application we’re thinking of is a little thing called Windows XP, which came out in October 2001. The “Dr. Watson” program in Windows goes back many versions, but it was in XP that it began helping people in real time. When a program stops working in Windows XP, Dr. Watson senses the problem, grabs the application crash data, and—if allowed—sends that information to Microsoft. And that data can change programs. Ben Canning, group program manager for Microsoft’s Office group, said a critical update for Office 2003 was released as a direct result of problems the company learned about in real time via Dr. Watson. “Late in the development cycle,” Canning recalls, “we made a change that prevented certain types of documents from opening because it looked like they caused a security problem. We got reports over a very short time that thousands and thousands of documents had been attacked in this way. That couldn’t have been happening. What we learned was that some files created in earlier versions of Office were setting off the security alert. We were able to create and distribute a fix quickly. Before, it would have taken time for that sort of problem to percolate through calls to the help desk. Now we can push a solution to the customer, because we know what the problem is much faster.”

It’s an article of faith among developers and other Web 2.0 true believers that one of the key differentiators between the new generation of hot Web 2.0 apps and the previous generation of dinosaurs is continuous iterative development. But if Microsoft, the poster child (poster grandparent?) for packaged software, can boast a service in which users of a program can drive continuous incremental improvement, this must be a concept that goes way beyond narrow definitions of Web 2.0.

---

All software development is iterative, of course. Perhaps what's most new about Web 2.0 is that this iterative development is made public quickly, and it happens on a server, not a client. Since a developer has to get an enhancement working on a local machine, not 100,000 remote machines, the development is smoother, faster, more reliable. As Google's Eric Schmidt said at the Web 2.0 Summit, "developing on the server is a fundamentally better model."

Developing on the server almost guarantees faster turnaround for improvements—and for fixing mistakes. Once eBay developers changed the auctioneer's popular yellow background color to white and the site's customers threw a collective fit. Realizing the error in moving too fast and not wanting to disturb site visitors any more than necessary, the developers of the site reverted the background color to yellow and then, over the course of months, very slightly lightened the background every day until it returned to the white. No fits that time.

## From nice-to-have to gotta-have

As Nathan Torkington noted a few pages ago, continuous iterative improvement is one of the key elements of successful Web 2.0 organizations. There is no more successful Web 2.0 organization than Google, and Marc Lucovsky is at the center of continuous iterative development there.

Since joining Google in 2005, Lucovsky has been directing development of the Ajax Search API, which lets developers sneak Google search into unexpected places (before moving to Google, Lucovsky was at Microsoft, where his projects included Windows NT and the ambitious Hailstorm initiative). Lucovsky says it's the "increased agility" of building a product iteratively that makes his job fun—even if it raises expectations of what he can accomplish. "People expect constant improvement now. They don't expect things to stand still. For my product right now we're still making changes a couple times a week. Most of the changes recently have not been super big or visible, but they are improvements."

The web, Lucovsky says, is open to new businesses and new approaches because of this increase in speed and expectation, both for web-based applications and those of the old-school variety. "It's a great distribution medium, great to help programs phone home to their developers, no matter what type of programs they are. Packaged software has long release cycles, sure, but there's nothing preventing packaged software from incorporating incremental continuous improvements. A good example of that is Windows Update. They're improvements, not radical changes, but they make programs more usable and more stable."

Lucovsky sees the emergence of new features across the web as examples of continuous iterative improvement in action. He says, "Look around the web, and you'll see that a lot of articles have a 'Digg this' icon. No one went in and designed a Digg button out of the gate. But people expect that when things get hot you deal with them and include them in your own products or websites. And you can do it without having to buy a Digg development kit." →

---

## Faster Than Fast

Google's Marc Lucovsky says that continuous iterative improvement means "we can add functionality that will have a ripple effect across the net instantly." Think he's exaggerating? Here's what happened the night of September 28, 2006 and the following morning:

**10:30 am:** A customer (using the handle "CyberCoder") working with the Ajax Search API asks if there's a way to change how video search is presented on a web page.

**10:42 am:** Mark replies that the customer could build a new feature starting with their code.

**10:52 am:** CyberCoder replies that he is not a hardcore JavaScript programmer, but "will give it a try anyway."

**around 9:00 pm:** In his home office, Mark "starts playing around" and develops the feature CyberCoder wants.

**around 11:00 pm:** Mark updates the Google site to include the new code and documentation.

**11:58 pm:** Mark writes a short blog post, announcing the launch of a "Video Bar"

At **3:29 am** the following morning (this is west coast time; CyberCoder lives on the east coast), CyberCoder notices Mark's post and, soon after, adds the code to his website. Mark writes: "The point of this timeline isn't to showcase how cool the feature is or isn't. The point is to show that in a world where platform development can happen in real time, engineers can produce a feature, a UI component, a new API. They can 'release' their work, and within hours their customers can consume their work and enhance their applications. The ability to do software development at these rates is game-changing. Developers get instant feedback, customers can provide ideas for incremental improvements and experience those improvements virtually instantly. Everyone, including the API authors, the programmers consuming their work, and the end users seeing the integrated solutions benefit from this vibrant and dynamic ecosystem. If we look at the key aspects of the next-generation computing platform that is the foundation of Web 2.0, this high-speed collaborative development environment is an important foundation."

For the full discussion thread, see [http://groups.google.com/group/Google-AJAX-Search-API/browse\\_frm/thread/4fe5cdbccee02012/40c32f760d955611?vc=1&fwc=1#](http://groups.google.com/group/Google-AJAX-Search-API/browse_frm/thread/4fe5cdbccee02012/40c32f760d955611?vc=1&fwc=1#)

**Continuously improved — and hard to do**

“Every writer about software sooner or later ends up offering a law under his own name,” Scott Rosenberg acknowledges. Here’s his inevitable addition to that canon:

**Rosenberg’s Law:**

Software is easy to make, except when you want it to do something new.

**Corollary:**

The only software that’s worth making is software that does something new.

“You’re going to get there much faster if you take small bites, move resolutely, learn, and move on.”

— Scott Rosenberg

→ Such improvements get integrated into the cost of doing business on the net. A “Digg this” button is yet another function (along with an RSS button, tags, trackbacks, and many more) that went from being cool to being necessary.

### Can software keep up with customer demands (or even book deadlines)?

Writing any kind of code is iterative, Scott Rosenberg notes in his recent *Dreaming in Code*, which chronicles his time as an embedded journalist inside the Open Source Application Foundation, Mitch Kapor’s project to create a breakthrough personal information manager. Rosenberg lets us in on the debates inside the OSAF: Should Chandler, as the PIM has been codenamed, be server-based? Should it be a web app? What features are essential and which can wait? Those debates have gone on so long that Rosenberg’s book, published via the usual glacial print-publishing schedule, has been published long before a completed Chandler is due. Rosenberg says, “You’re going to get there much faster if you take small bites, move resolutely, learn, and move on. That was their goal.”

Hindsight, unavailable when the project began, suggests that Chandler might have gotten to market faster as an inside-the-browser application. Rosenberg guesses that, “For Chandler to succeed, we have to wait for the pendulum to swing back from web apps. And there are some disadvantages to the web app approach. One that concerned Mitch early on was that you have to have network connectivity for it to be of value. That’s less of an issue now than it was in 2002, but it’s still an issue. Also, unlike the old desktop days, Web 2.0 is an invitation for malware and bad actors. When you send data over the net, that opens a possibility for bad things to happen. It’s a small possibility but still. We’re throwing all our lives on the web. Someday we’re going to go too far and regret it.”

### Your life is on the web. Now what?

More and more of us are, indeed, throwing our lives on the web: storing our email and contacts in Gmail, our personal financial information in Wesabe, and our resumes in LinkedIn. A dozen years after Amazon.com opened its virtual doors, our credit card numbers reside on countless servers across the net. We hope they’re secure.

It’s a risk we’re willing to take. Why? Because, despite many legitimate security and privacy concerns, having our data on the web is what facilitates continuous iterative improvement. Letting Google aggregate the information it finds in browser toolbars makes PageRank more accurate. Recommendation engines from iTunes and Last.fm become more relevant based when they have more of your more data at their disposal. Weather, traffic, and movie time reports are more specific because you shared some geographical information. These are all services in which the data being input changes frequently and the data being output changes frequently.

---

The amount of code being changed on popular web sites in short periods of time is impressive: Cal Henderson says that, during its initial development, Flickr pushed new versions every half hour. On a slightly more relaxed pace but a much larger scale, eBay publishes a new version of its site every two weeks, massaging more than 100,000 lines of code—while the site keeps running. The very notion of software version releases may be changing. Marc Hedlund, chief product officer of Wesabe (and a former entrepreneur in residence at O'Reilly) suggests developers get out of the version number mindset and into the timestamp mindset (see [http://radar.oreilly.com/archives/2006/02/web\\_development\\_20.html](http://radar.oreilly.com/archives/2006/02/web_development_20.html)). Just hope that the 4:47 version doesn't reintroduce the problem introduced by the 12:38 version that you thought had been solved by the 2:11 version.

In this context, the future is going to look a lot like the music recommendation engine Last.fm. We'll see more and more small, efficient, mobile, fast-changing services that draw on and add value to existing stores of data (while creating more data for others to exploit). Just around the corner from the mainstream are many more examples of services that "scrobble" the net data that used to exist only locally. Imity and Jaiku stretch ideas regarding what you can get out of a mobile phone, Kayak and other mobile providers aggregate and redistribute travel data, Joost and other projects distribute everyone's TV data everywhere, MyBlogLog helps you define yourself to the world as the sum of all the sites you visit. Who you are, at least from the point of view of the data you create, keeps changing. Web 2.0, as it turns out, is the way to keep up with that.

It's not all about individuals, though. Continuous iterative development isn't just for the consumer internet. Forward-looking corporate IT departments who want to differentiate themselves from the bloated IT infrastructures and development policies of recent decades will want to look at this approach to create competitive differentiation for their companies. If your company doesn't set up a development process in which new iterative versions follow one another quickly, your competitors just might.

There's a data angle here, too. While practicing continuous iterative improvement, companies need to understand what data they have collected, and how to engage their customers in enhancing it. One of the best ways to do that is to figure out how to give users the benefit of that data. A credit card company could help people understand their spending patterns. A phone company could create a networked address book of people customers call regularly. Data mining shouldn't be some periodic back office activity. It needs to be conducted regularly and brought "up front," into the interface: where a company's customers are and where frequent changes can be made apparent quickly. Superfast changes, after all, aren't only for a company's developers—they're for a company's customers. ■

---

### Continuous iterative improvement didn't start with Web 2.0

The Japanese term "kaizen" (kai means "change" and zen means "good") goes back to the 1950's. According to Wikipedia, it's "an approach to productivity improvement originating in applications of the work of American experts...and of the War Department's Training Within Industry program by Japanese manufacturers after World War II. The development of Kaizen went hand-in-hand with that of quality control circles, but it was not limited to quality assurance. The goals of kaizen include the elimination of waste, just-in-time delivery, production load leveling of amount and types, standardized work, paced moving lines, right-sized equipment, etc. ... A closer definition of the Japanese usage of Kaizen is 'to take it apart and put back together in a better way: What is taken apart is usually a process, system, product, or service.'

The most famous corporate use of kaizen is at Toyota, where line personnel are encouraged to stop production when any problem appears. There are incentives for improving products in process. It's a practice relevant to the Web 2.0 world—continuous iterative development is another reason why Toyota is moving in one direction and the more hierarchical automakers Ford and General Motors are moving in the other.

### Get real

Looking for best practices on continuous iterative development? We recommend 37signals' ebook *Getting Real*.

See <http://gettingreal.37signals.com>

# The Number: One Person Per Blog

Each issue we will look at one statistic in business and technology and explore what is happening behind that number. We begin by considering a piece of data as delivered by the head of a company that knows a thing or two about data.

*It's a joke, it's not quite true, and those who believe it are missing an opportunity*

by Jimmy Guterman



Courtesy of Google Video

Found on:

<http://video.google.com/videoplay?docid=5625662279952157836&q=Eric+Schmidt%27s+Speech+to+Conservative+Party+Conference+2006>.

“The average blog has exactly one reader: the blogger.”

— Eric Schmidt

On October 2, Google CEO Eric Schmidt spoke to the leaders of England’s Conservative Party. He ran through a series of data points uncovered by search engine, then slowed down and peaked with “We have 35 million blogs, doubling every six months. The average blog has exactly one reader: the blogger.” The news was greeted by laughter, then applause. Schmidt went on, “It’s fine. There are a lot of very successful blogs and there are an awful lot that are not.” The line about one reader per blog may now be part of Schmidt’s stump speech: he repeated it, to a similar response, the next month stateside at the Republican Governors Association meeting.

Schmidt is an extremely intelligent person who knows the nuances of the Net better than almost anyone, but what he’s purveying in these talks is the conventional wisdom: bloggers are talking to themselves. Michael Kinsley writes for *TIME*, which is about as close as you can get to the epicenter of conventional wisdom, even at this late date in the life of general-news print magazines. In a recent column mulling over whether newspapers have a future, he went after “the blog terror: people are getting their understanding of the world from random lunatics riffing in their underwear, rather than professional journalists with standards and passports.” That sentence can be torn apart in a dozen ways, but briefly: there’s no evidence, even from boosters, that even a small percentage of people are “getting their understanding of the world from blogs.” I know plenty of bloggers who wear more than underwear; a passport is no guarantee of insight; and there are plenty of “random lunatics” broadcasting in all media, not just blogs. Indeed, for many years Kinsley was host of a CNN show, *Crossfire*, that rarely featured reasoned, sober discourse. Yet the myth persists that the bloggers are the problem.

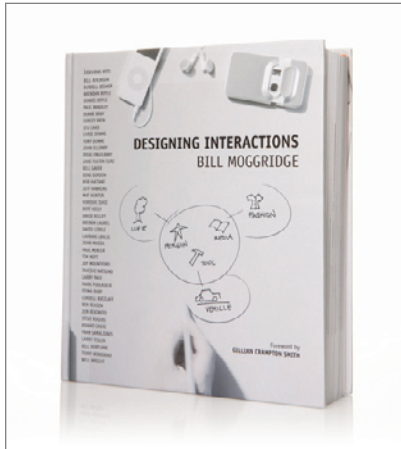
The truth is, at the top of the blog charts, you’re inundated by mainstream voices from mainstream media: voices like Kinsley’s. According to the most recent survey by Technorati, the blog search engine, 47 of the top 50 blogs are associated with mainstream media such as *The Washington Post* and the BBC. Even if Kinsley’s nightmare comes true and blogs become the way most of us understand the world, most of those bloggers at the head of Chris Anderson’s “long tail” are as mainstream as the op-ed section of big daily newspaper. Indeed, some of those popular bloggers are op-ed columnists for such newspapers.

But the short head of the ever-lengthening long tail tells only part of the story. As with so many forms of media, especially young and unstable ones, what’s interesting is not necessarily what’s happening at the top. As you work your way down the long tail of more than 60 million public blogs, of course you work your way through to many blogs (perhaps in the hundreds of thousands, likely higher) written for few people or no one. As Technorati’s David Sifry notes, “The further out you go on the long tail, because of the frictionlessness of the media, it’s very easy to create a new blog. You can create one for every shopping trip you go on, one just for your bowling league, one just for your school board. But forget about the bottom of the tail. There’s enormous power in the



# The Canon:

So much of what we have to read for business—an email, a report, an agenda—is extremely important right this moment, but will be less and less so as time passes. In a fast-moving industry, that's inevitable. But there are ideas that are meant to last for generations. In each installment of "The Canon," we'll look at one book, article, website, or whatever else we can find that offers ideas and approaches that will last and are essential to mastering today's technology business—and tomorrow's.



Sometimes the key to a successful interactive product is eliminating the elements that serve to confuse. photo: Nicolas Zurcher

## *Designing Interactions*

Bill Moggridge

The MIT Press

<http://www.designinginteractions.com>

by Jimmy Guterman

Most of what we will cover in "The Canon" will be material that has proven its value over time, but we're starting with a book/DVD combo that came out only last year but feels like an instant classic.

There are many wonderful books about design and its implications for business and technology, ranging from the extremely accessible (Donald Norman's *The Invisible Computer*) to ideal for practitioners (Jakob Nielsen et al's *E-Commerce User Experience*) to the joyfully excessive (Gavriel Salvendy's *Handbook of Human Factors and Ergonomics*, all 1,654 pages of it). We recommend them all, but *Designing Interactions*, by Bill Moggridge, a founder of the IDEO design firm, distinguishes itself by offering all sorts of practical tips for what it takes to make breakthrough products and services, and embedding those lessons inside stories of the people who made them happen.

The book is built on interviews with 40 innovative and influential designers, among them pioneers of computers and handhelds, all sorts of telecommunications, games, and notions that still feel more futuristic than real. In this big book stuffed with useful information, Moggridge is comfortable slowing down and letting us see through his idiosyncratic eyes. When he describes the circa-1981 Osborne luggable computer, he compares it when it's closed to a sewing machine, when it's open to a human face, and the keyboard cable looks "like the tongue of a cheeky little boy." A quarter century after he first saw the object, Moggridge's wonder is still very much alive and infectious.

The practical lessons are there, don't worry. But it's the stories that will keep you reading: Larry Tesler and Tim Mott working tag-team around the clock at Xerox PARC to control access to their precious Alto; Bill Atkinson inventing pull-down menus during one mysterious night; Brendan Boyle using memories of Charlie Brown and Lucy to create his Aerobie Football. Nearly all the interviewees reach beyond the expected when they describe what they've achieved: Bing Gordon of Electronic Arts talks of designing games and says "we imagined that people using technology could have the same kind of fun that lion cubs have when they're tussling in the Savannah."

Most book/DVD combos are gimmicky, but here's one case in which having a DVD with excerpts from 37 of the interviews increases your enjoyment and improves your understanding. Seeing and hearing these masters discuss their craft reinforces the lessons inside the book—and it lets you get through the whole thing in less than two hours if you're not committed to completing an 800-page book. Like so much of the outstanding design it celebrates, *Designing Interactions* lets you use it the way you want to.

Behind the business and design insights, there's a life lesson here as well. Although written by a legend in the field, *Designing Interactions* turns out to be a great case study in humility. Not until the final chapter does Moggridge deliver his own maxims, each one more surprising and spot-on than before. Until then, he does what the smartest designers and businesspeople do: he listens and he learns. Only then can you develop the ideas that will build innovative businesses. ■

---

# Coming soon

If you know of any good examples of the categories listed below, please let us know.

- Next Issue: Sure, code is law, but which businesspeople and alpha geeks are really doing their work in ways that take into account this profound maxim?
- The elements of digital style (apologies to Strunk and White)
- Microsoft, open source, and some red-state/blue-state surprises
- The end of simplicity
- Web 2.0 and Big Business
- The videogame console wars (and which numbers to believe)
- Open source finance
- Why your website still stinks
- Digital rights management is dead. Now what?
- Open businesses, proprietary advantages
- Software as storytelling
- How Apple gets away with it
- Do you really want to hand over control to your customers?
- Visualizations
- The limits of efficiency
- How to open closed systems (yes, we're thinking about mobile networks)
- Predictive markets
- Wikipedia and its children (or, community publishing and its discontents)
- Discovering the web's edge
- Identity
- And much more...

# Calendar

A selection of significant events over the next few months.

**T** Events Tim O'Reilly plans to attend.

**D** Events Jimmy Guterman plans to attend.

Lack of a symbol is no indication of lack of merit. Please contact Jimmy (jimmy@guterman.com) to let us know about other events we should include.

February 24

- D** **Beyond Broadcast** (Cambridge, MA) <http://beyondbroadcast.net>  
From participatory culture to participatory democracy, attendees will explore the means, the message, and the meaning of the post-midterm, pre-presidential YouTube moment.

February 27–March 1

- T** **O'Reilly Emerging Telephony Conference** (Burlingame, CA) <http://oreillynet.com/etel2007/>  
This conference explores the strategies for taming disruption and exploiting opportunities being created by web telephony innovations. conferences.

March 5–6

**Freedom to Connect** (Washington, DC) <http://freedom-to-connect.net>  
How networks are changing our fundamental economic and social assumptions.

March 7–10

- D** **TED** (Monterey, CA) <http://ted.com> (see also <http://ted.com/tedglobal2007/>)  
No longer a big celebration revolving around Richard Saul Wurman, Chris Anderson's conference still covers the fertile ground where technology, entertainment, and design meet.

March 9–13

**South By Southwest Interactive** (Austin, TX) <http://2007.sxsw.com/interactive/>  
Be part of the event that seems to double Austin's population for a week. This year the interactive part of the multimedia conference promises "a full range of geek-related topics."

March 26–29

- T** **D** **O'Reilly Emerging Technology Conference (ETech)** (San Diego, CA) <http://conferences.oreillynet.com/et2007/>  
What technological rejiggering or change in perspective is poised to blast off into the realm of magic? This conference casts a wide net aiming to answer that question. From the technological infrastructure supporting mass-market players to the promise of alternative energy sources, ETech looks at what's here while keeping an eye firmly trained on what's coming.

April 5

**Ignite!** (Seattle, WA) Make contests and tech talks. <http://igniteseattle.com>

April 15–18

- T** **D** **Web 2.0 Expo** (San Francisco, CA) <http://web2expo.com>  
Experience 30,000 square feet of Web 2.0, sponsored by O'Reilly and CMP.

- 
- April 26
- **The Economics of Social Media** (Los Angeles, CA) <http://econsm.com>  
Sponsored by paidContent, mocoNews, and contentSutra, this bills itself as “the first conference focused on business models and deals as much as the creative process and enabling technologies in the social media ecosystem.”  
Disclosure: This newsletter’s editor is the conference’s programming chair.
- May 19–20
- **Maker Faire** (San Mateo, CA) <http://makerfaire.com>  
Sponsored by O’Reilly’s *MAKE* and *CRAFT*. Come for a weekend of projects, eye-opening demonstrations, and fun. Build, craft, hack, play, make.
- May 29–30
- **Where 2.0** (San Jose, CA) <http://conferences.oreillynet.com/where2007/>  
The third annual O’Reilly-sponsored conference brings together the people, projects, and issues building the new technological foundations and creating value in the location industry.
- May 30–June 1
- ■ **D: All Things Digital** (Carlsbad, CA) <http://d.wsj.com>  
Sponsored by *The Wall Street Journal* and hosted by Walt Mossberg and Kara Swisher.
- June 18–20
- **O’Reilly Tools of Change for Publishing Conference** (San Jose, CA) <http://conferences.oreillynet.com/toc/>  
This debut conference intends to raise the level of technology knowledge and discourse in the publishing industry and provide a meeting ground for those leading the charge into the future of publishing.
- June 20–22
- **Flight School** (Aspen, CO) <http://edventure.com>  
Esther Dyson doesn’t edit this newsletter anymore, but she does continue to host this conference on the startup airline market, with a focus on air taxis.
- June 20–22
- **Supernova 2007** (San Francisco, CA) <http://supernova2007.com>  
Supernova is where the superconnectors of the IT industry connect. Organized by another former *Release 1.0* editor, Kevin Werbach, and produced in partnership with The Wharton School.
- January 14–18
- **Macworld Conference and Expo** (San Francisco, CA) <http://macworldexpo.com>  
We have no idea what will be introduced at that event, but we do suspect the desire for it will be intense and irrational.

---

# Release 2.0 Subscription Form

Complete this form and join the other industry executives who rely on *Release 2.0* to stay ahead of the headlines. Or, if you wish, you can also subscribe online at [r2.oreilly.com](http://r2.oreilly.com).

Your annual *Release 2.0* subscription costs \$495 per year, and includes both the print and electronic versions of six every-other-month issues, access to the complete *Release 1.0* and *Release 2.0* archives, a discount on attendance to O'Reilly conferences, and full access to our website, <http://r2.oreilly.com>.

name \_\_\_\_\_

title \_\_\_\_\_ company \_\_\_\_\_

address \_\_\_\_\_

city \_\_\_\_\_ state \_\_\_\_\_ zip \_\_\_\_\_ country \_\_\_\_\_

telephone \_\_\_\_\_ fax \_\_\_\_\_

email (personal email required for electronic access) \_\_\_\_\_

url \_\_\_\_\_

My colleagues should read *Release 2.0*, too!  
Send me information about multiple copy subscriptions and electronic site licenses.

check enclosed     charge my (check one)     american express     mastercard     visa

card number \_\_\_\_\_ expiration date \_\_\_\_\_ cvs code \_\_\_\_\_

**name and billing address**     same as above     see below

name \_\_\_\_\_

address \_\_\_\_\_

city \_\_\_\_\_ state \_\_\_\_\_ zip \_\_\_\_\_ country \_\_\_\_\_

signature \_\_\_\_\_

Please fax this form to 1.818.487.4501 or mail it to Release 2.0, PO box 17046, North Hollywood CA 91615-9588

Payment must be included with this form. Your satisfaction is guaranteed or your money back.

If you have any questions, please call us at 1.800.889.8969 or 1.707.827.7019 or email us at [r2.oreilly.com](mailto:r2.oreilly.com).

