Beautiful Software

Ed Yourdon
email: ed@yourdon.com
Website: www.yourdon.com Blog: www.yourdonreport.com
Twitter, LinkedIn, Facebook, Plaxo, Flickr: "yourdon"

NYC SPIN presentation
Sep 21 2010

blue stuff = "external" links — i.e., URLs to pages on the Internet

Publication Details, and General Disclaimer

- This presentation is an open-content collaborative document. Anyone with an Internet connection and World Wide Web browser may view and/or alter its content — for better or worse. Please be advised that while the material in this presentation has been reviewed by Ed Yourdon ("Ed"); the theories and business practices expressed by the document are not necessarily his.

- This isn't to say you won't find valuable and accurate information herein; however, Ed cannot summarily guarantee the validity of this document. The content of any given page may recently have been changed, dumbed-down, or otherwise edited by someone whose opinion does not correspond to Ed's original material (or any subsequent drafts).

- Neither Ed, nor any of the contributors, collaborators, nor anyone else connected with this document, can in any way whatsoever be held responsible for the appearance of any inaccurate information, or for your use of the information contained in or linked from this document.

- You are being granted a limited license to copy anything from this document; it does not create or imply any contractual or extra-contractual liability on the part of Ed, nor any of the contributors, collaborators, or viewers of this material.

- There is no agreement or understanding between you and Ed regarding your use or modification of this information beyond the GNU Free Documentation License (GFDL); neither is Ed responsible should someone change, edit, modify, or remove any information that you may post on this document.

- Any of the trademarks, service marks, collective marks, design rights, personality rights, or similar rights that are mentioned, used, or cited in this document are the property of their respective owners. Their use here does not imply that you may use them for any purpose other than for the same or similar informational use — as recognized under the GFDL licensing scheme. Unless otherwise stated, Ed and this document are neither endorsed by nor affiliated with any of the holders of any such rights; as such, Ed cannot grant any rights to use any otherwise protected materials. Your use of any such or similar incorporated property is at your own risk.

Beautiful Software

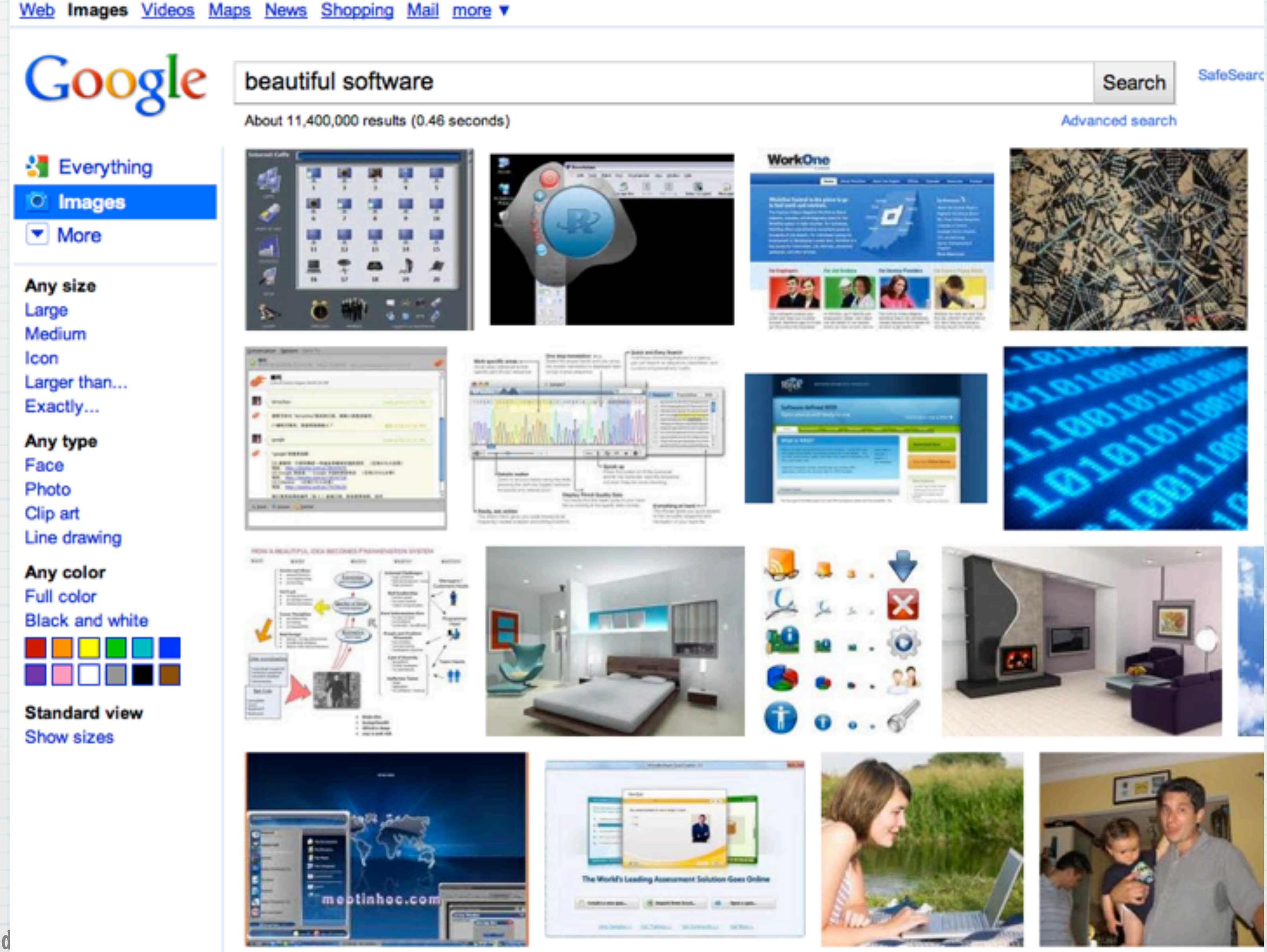
...

Huh?

Why don't you just f--ing google it?

Why don't you just f--ing Bing it?

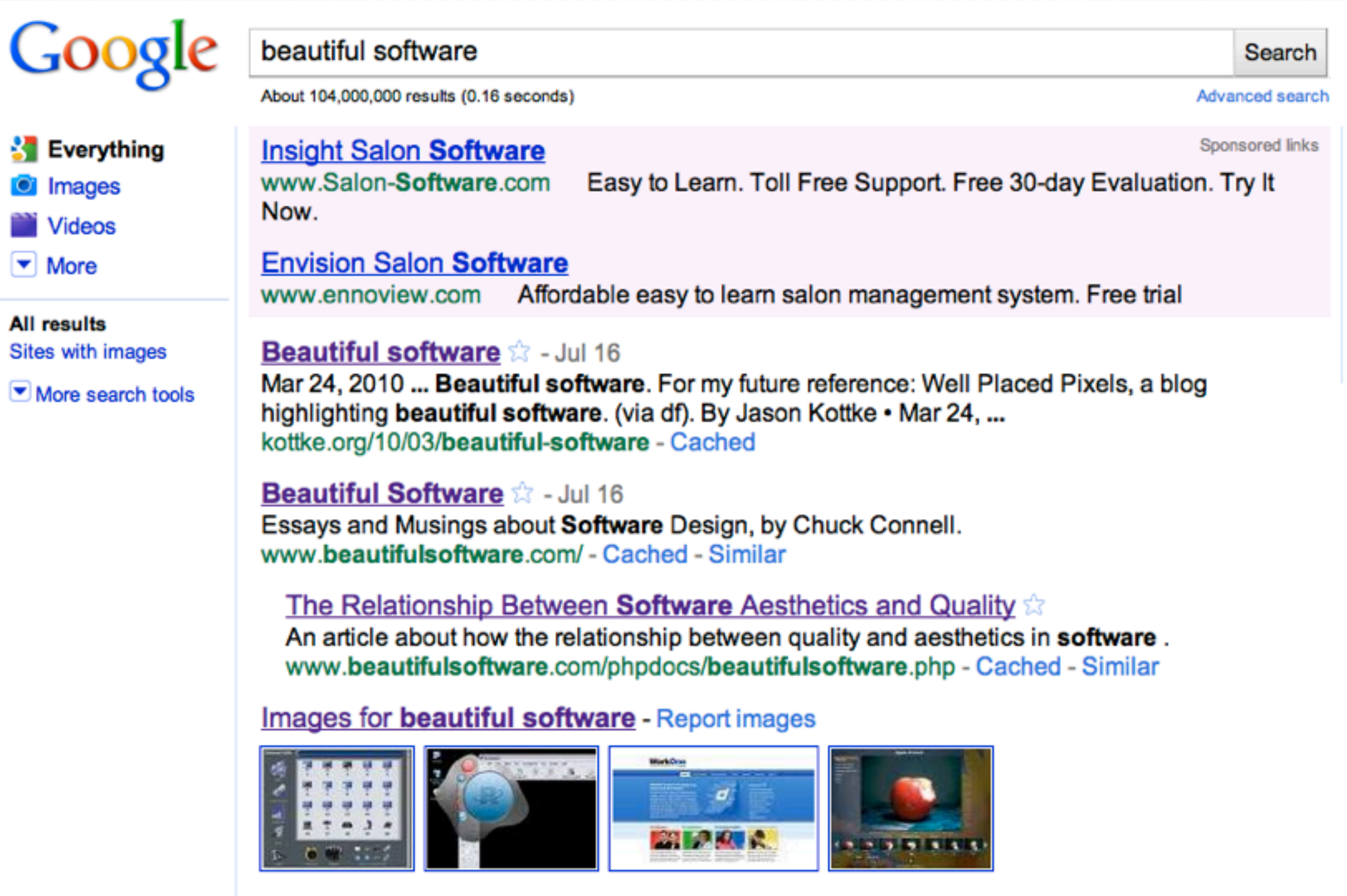
Google's take on beautiful software



beautiful software

About 11,400,000 results (0.46 seconds)

Google's take on beautiful software



Google

beautiful software

Search

About 104,000,000 results (0.16 seconds)

Advanced search

- Everything
- Images
- Videos
- More

All results

Sites with images

More search tools

Sponsored links

[Insight Salon **Software**](#)
www.Salon-**Software**.com Easy to Learn. Toll Free Support. Free 30-day Evaluation. Try It Now.

[Envision Salon **Software**](#)
www.ennoview.com Affordable easy to learn salon management system. Free trial

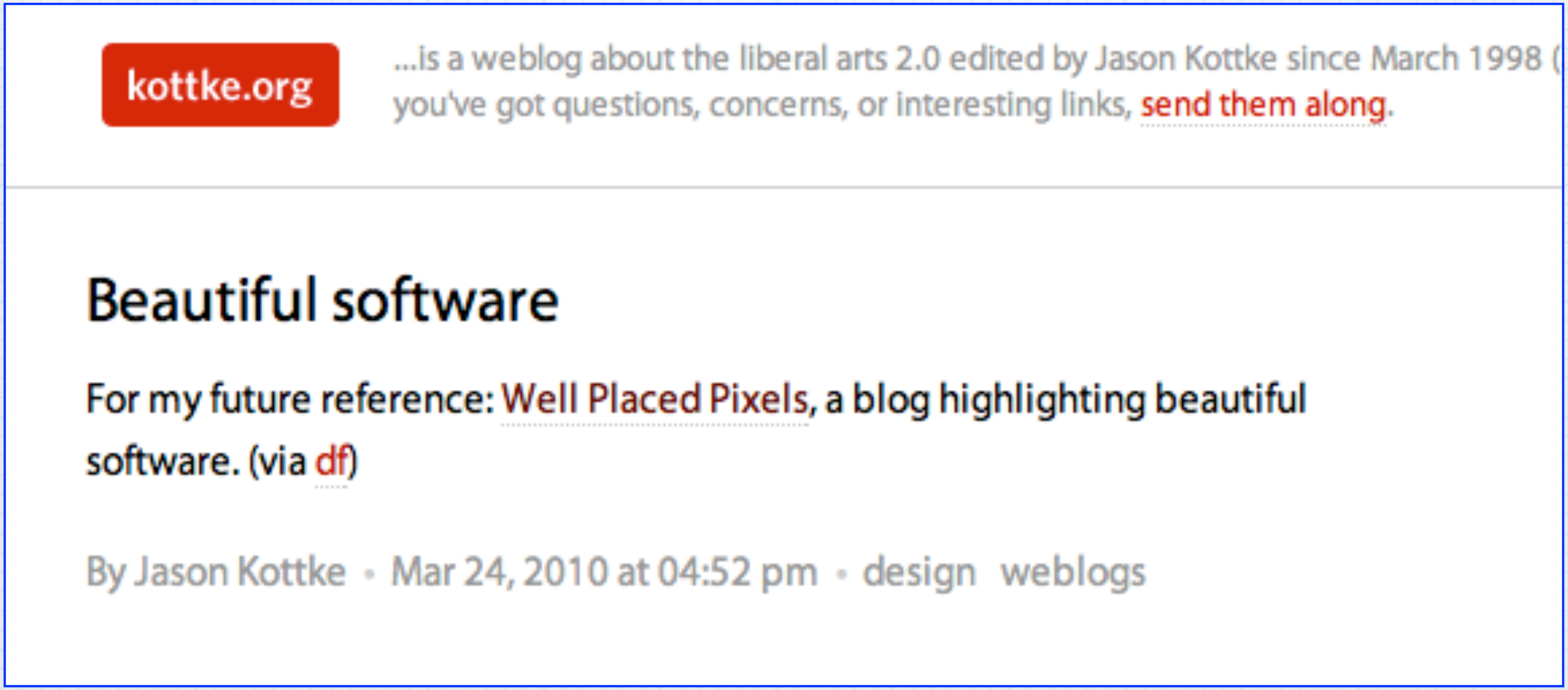
[**Beautiful software**](#) ☆ - Jul 16
Mar 24, 2010 ... **Beautiful software**. For my future reference: Well Placed Pixels, a blog highlighting **beautiful software**. (via df). By Jason Kottke • Mar 24, ...
kottke.org/10/03/**beautiful-software** - Cached

[**Beautiful Software**](#) ☆ - Jul 16
Essays and Musings about **Software** Design, by Chuck Connell.
www.**beautifulsoftware**.com/ - Cached - Similar

[The Relationship Between **Software** Aesthetics and Quality](#) ☆
An article about how the relationship between quality and aesthetics in **software** .
www.**beautifulsoftware**.com/phpdocs/**beautifulsoftware**.php - Cached - Similar

[Images for **beautiful software**](#) - Report images

Jason Kottke's web site



kottke.org

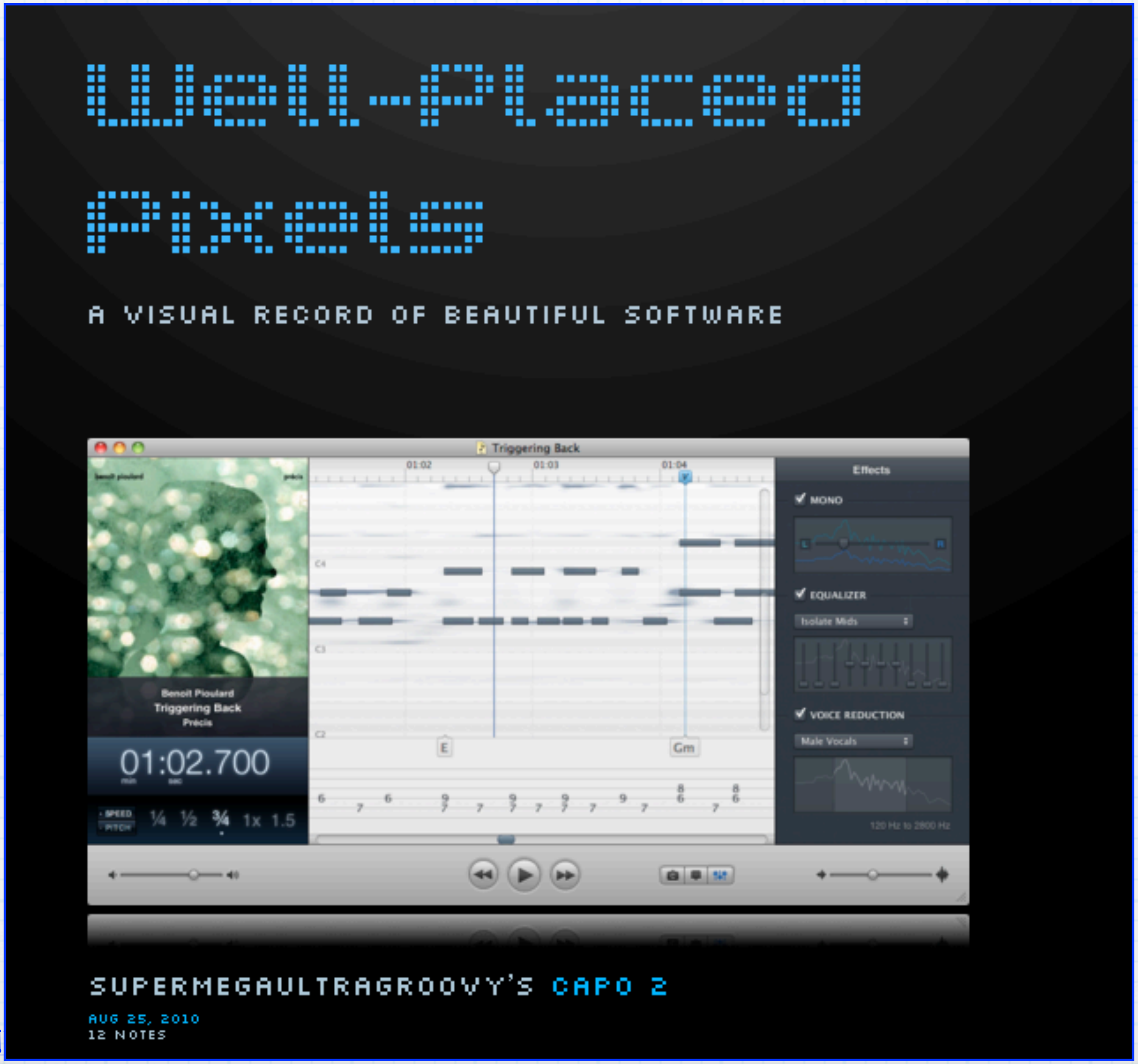
...is a weblog about the liberal arts 2.0 edited by Jason Kottke since March 1998 (
you've got questions, concerns, or interesting links, send them along.

Beautiful software

For my future reference: Well Placed Pixels, a blog highlighting beautiful software. (via df)

By Jason Kottke • Mar 24, 2010 at 04:52 pm • design weblogs

Well-placed pixels



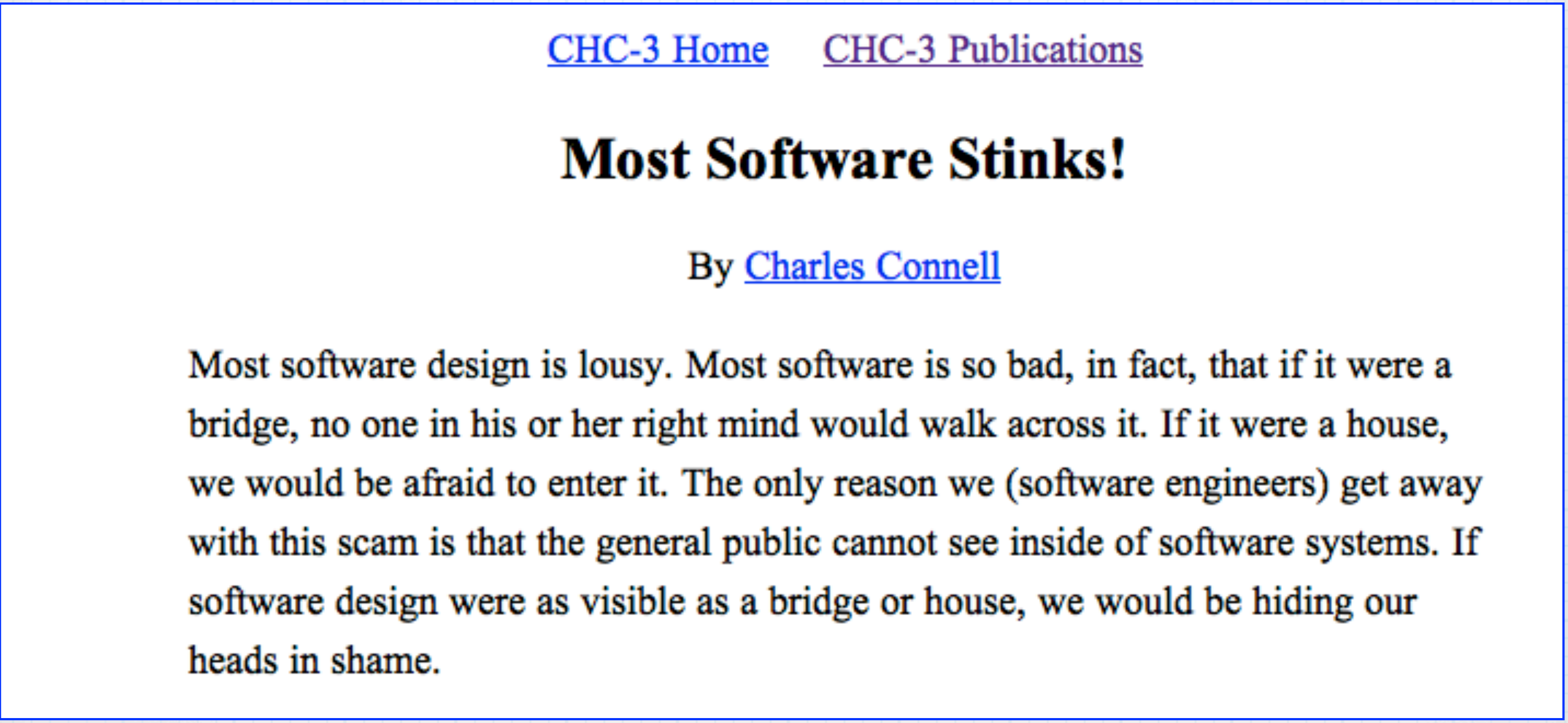
Well-Placed Pixels

A VISUAL RECORD OF BEAUTIFUL SOFTWARE

SUPERMEGAULTRAGROOVY'S CAPO 2

AUG 25, 2010
12 NOTES

Chuck Connell’s “Beautiful Software”



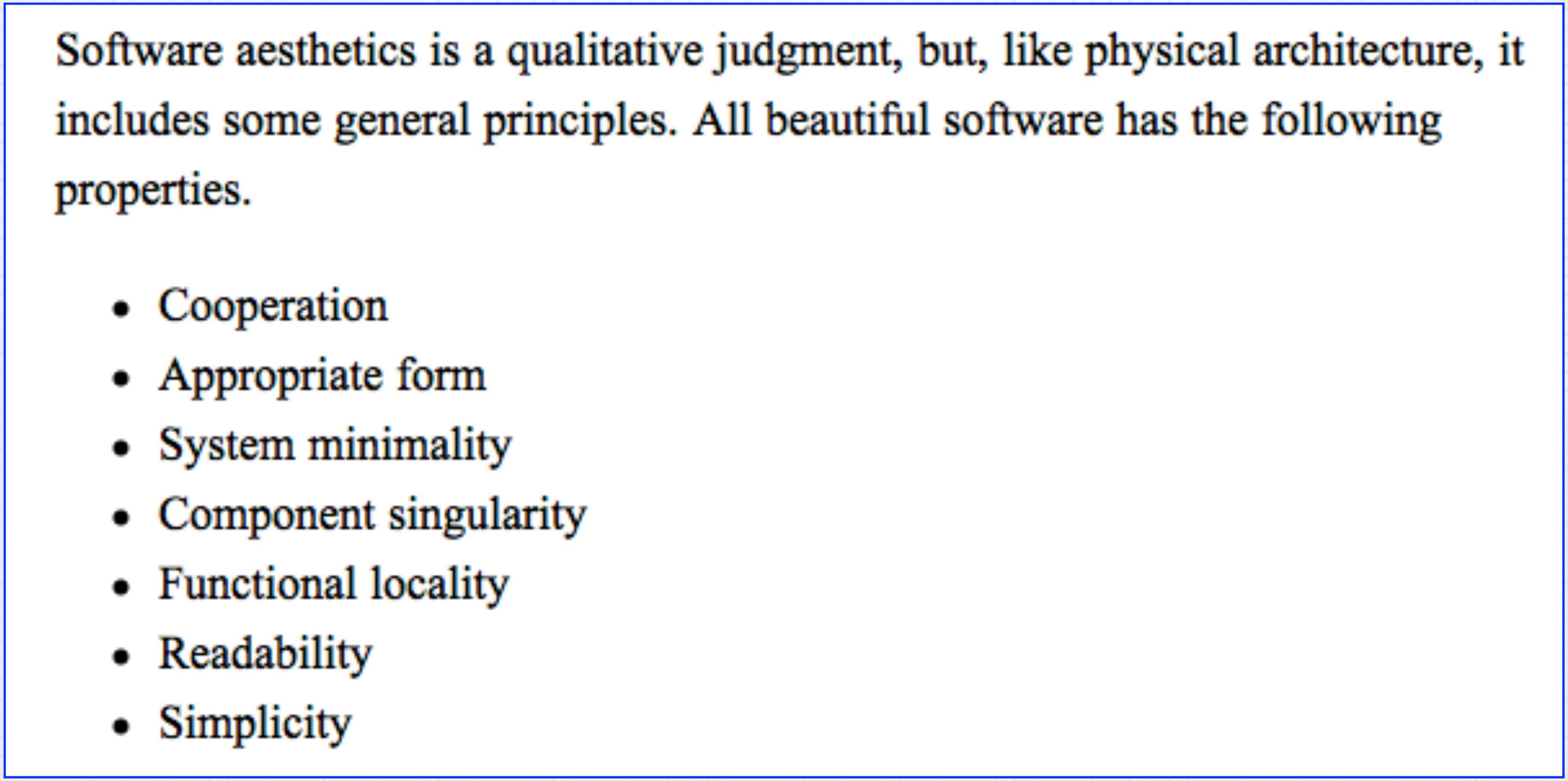
CHC-3 Home CHC-3 Publications

Most Software Stinks!

By Charles Connell

Most software design is lousy. Most software is so bad, in fact, that if it were a bridge, no one in his or her right mind would walk across it. If it were a house, we would be afraid to enter it. The only reason we (software engineers) get away with this scam is that the general public cannot see inside of software systems. If software design were as visible as a bridge or house, we would be hiding our heads in shame.

Chuck Connell's "Beautiful Software"



Software aesthetics is a qualitative judgment, but, like physical architecture, it includes some general principles. All beautiful software has the following properties.

- Cooperation
- Appropriate form
- System minimality
- Component singularity
- Functional locality
- Readability
- Simplicity

Beautiful software = quality?

- Traditional definition of quality
  - ☑ Fulfills the user's needs (which may or may not have been fully articulated in a formal requirements doc)
  - ☑ Delivered on time, or ahead of schedule
  - ☑ Developed within budget
  - ☑ No "show-stopper" bugs, and very, very few minor bugs
  - ☑ ... and lots of "ilities" (e.g. portability)

A journalist's perspective

- What?
- What not?
- Why?
- Why not?
- Who?
- How?

What

- Makes users gasp with delight every time they use it
- Makes them mutter, "this #!&*% software is **AWESOME**!"
- Note: legendary software guru **Eddie Izzard** says awesome is "like a hundred billion hot dogs"

Another perspective

- From "Five Star Apps Are Awesome, Simple, and Done" (New York Times, Sep 20, 2010)
  - ☑ Surveyed common words in 1-star and 5-star reviews of top 100 apps in iTunes App Store
  - ☑ Underlined words are likely to be used as adjectives in the comment
- Five Star Apps
  - ☑ Awesome, worth, thanks, amazing, simple, perfect, price, everything, ever, must, ipod, before, found, store, never, recommend, done, take, always, touch
- One Star Apps
  - ☑ waste, money, crashes, tried, useless, nothing, paid, open, deleted, downloaded, didn't, says, stupid, anything, actually, bought, apple, already

What not?

- “Antithesis: “ugly” software
  - ☑ Software that users hate every time they use it
  - ☑ Software they use only because they have no choice, no alternative (beware: they will find an alternative, sooner or later)
  - ☑ Often manifested by terrible UX/UI…
  - ☑ … but might also reflect inconsistent/inadequate functionality, and/or slow/buggy behavior
  - ☑ For more details, see Eddie Izzard’s YouTube encore on computers (warning: not quite safe for the office…)

Additional thoughts

- Like pornography: even if I can't describe ugly software, I know it when I see it
- However, what's beautiful to me might not be beautiful to you
- And software that was considered beautiful 20 years ago might not be beautiful today
- Question: is it the idea, or the implementation of the idea that matters?

Ugly Software

Examples of beautiful SW

- Google Maps
- Google (but not YouTube)
- Wikipedia (the idea, plus the full audit trail)
- CraigsList
- Visicalc
- High Volume Trading (HVT) systems
- Linux
- iPhone apps?
- Many systems & projects presented at PopTech

Why build beautiful software?

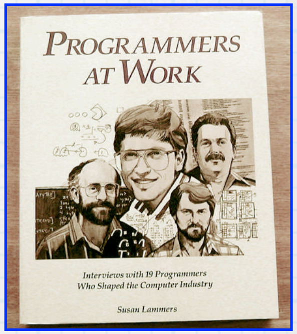
- Not because of slogans
- Because we want to change the world (but don't forget "Crossing the Chasm" and "Structure of Scientific Revolutions"
- Because even though we are engineers, we are also artists
- Because it makes us smile
- Because it makes us proud
- Because it makes us remember why we stumbled into this field in the first place
- Because it's something we can show off
- Related question: who are we doing it for? Ourselves? Our boss? Our users/customers?

Why not?

- They don't pay me to make beautiful software
- Nobody here cares about beautiful software
- That's not what we're in business to do
- I just want to make gobs of $$$ — I'll make ugly software if it pays enough
- None of this matters. It just pays the rent. My real passion is outside the office.

Who builds beautiful software?

- Sam Palmisano?
- Mark Hurd?
- Steve Jobs?
- Craig Newmark?
- Dan Bricklin?
- Jimmy Wales?
- Marc Andreesen?
- Sergei Brin and Larry Page?
- Thomas & John Knoll?



PROGRAMMERS AT WORK

Interviews with 19 Programmers Who Shaped the Computer Industry

Susan Lammers

How to build beautiful software

(suggestions that should be freely ignored!)

- Find a passion
- Find a nurturing environment
- Find advisors, mentors (young kids, tribal elders, etc.)
- Remain open — but don't compromise
- Be prepared to break the rules. Lots of rules.
- Be humble — adapt what's already out there
- Use an agile development approach
- Be prepared to collaborate
- Simplify, simplify, simplify
- Chances are that UI will be 90% of your success

Find a passion

- You should have a strong passion about something specific before you even begin trying to create it as a software-related beautiful thing
- But your job, company, friends, spouse, and surroundings may be distinctly non-passionate
- Find passionate people who can inspire you
- Look outside your job/company/field, if necessary — art, music, or any other human endeavor
- Look for places where you can find inspiration (e.g., PopTech, NY Tech Meetup)

FInd a nurturing environment

- It might be a university or R&D shop of some kind
- Or a "pocket of excellence" inside a traditional shop
- Find someplace that has a policy similar to Google's "20 percent time"
- A variation: take a boring, low-stress job to pay the rent; write beautiful software on your own time.
- The software doesn't have to be your own "invention" — it could be an open-source project
- The Web 2.0 paradigm: you don't need VC's any more.
- Remember: you can collaborate with people remotely

Find advisors, mentors

- These may be tribal elders
- Or young kids — e.g., university students you can hire as cheap (or free) interns
- Or kids out on the street
- Or a “brain trust” of individuals you’ve met or worked with in the past
- You want people who will give you honest, realistic feedback ...
- ... but who won’t be “dream-killers” (as well-meaning friends and spouses sometimes are)

Remain open — but don't compromise

- This advice seems intrinsically contradictory, but really isn't
- You must not compromise your passion, or your determination to make something beautiful
- But you need to be "open" to new tactics & strategies
- And new "purposes" for your beautiful creation (inventors are usually the ones least able to see how their creation will actually be used)

Break the rules

- Be prepared to break the rules. Lots of rules.
- This is one reason that you'll probably have to develop your beautiful software outside the traditional corporate environment.
- Especially be prepared to break administrative and bureaucratic rules
- But also be prepared to break social/cultural rules — because your younger user/customers will be doing that already, as part of their normal spirit of rebellion
- Look for technical rules that you can break (and remember Moore's Law)

Be humble — adapt what's there

- Reinventing the wheel is usually a waste of time
- And others may have already developed brilliant solutions to specific parts of your creation — which are available as open-source solutions.
- A unique combination of ordinary, mundane parts can set the world on fire (and can also be patented, and can be legitimately claimed as a trade secret)

Use an agile approach

- Duh.
- If you need to be persuaded about this, you should probably be in a different line of work...

Be prepared to collaborate

Simplify, simplify, simplify

Chances are UI/UX will be 90% of your success

- Most of us have seen lots of examples of bad UI design
- ... and perhaps a few good ones, too
- But most of us are amateurs in this area
- And many of us are brainwashed by the UI's we grew up with (e.g., traditional GUI windows)
- We can read books (e.g., Edward Tufte's **Visualizing Information**), take courses, etc...
- Data mining, data visualization likely to be big issues for future products
- Look at interesting research areas (e.g., **Digg Labs**)
- Look for new UI paradigms (touch, speech, "ambient" devices, etc.)

A journalist's perspective

- What?
- What not?
- Why?
- Why not?
- Who?
- How?

Summary & Conclusions

Beautiful Software

Ed Yourdon
email: ed@yourdon.com
Website: www.yourdon.com Blog: www.yourdonreport.com
Twitter, LinkedIn, Facebook, Plaxo, Flickr: "yourdon"

NYC SPIN presentation
Sep 21 2010

blue stuff = "external" links — i.e., URLs to pages on the Internet