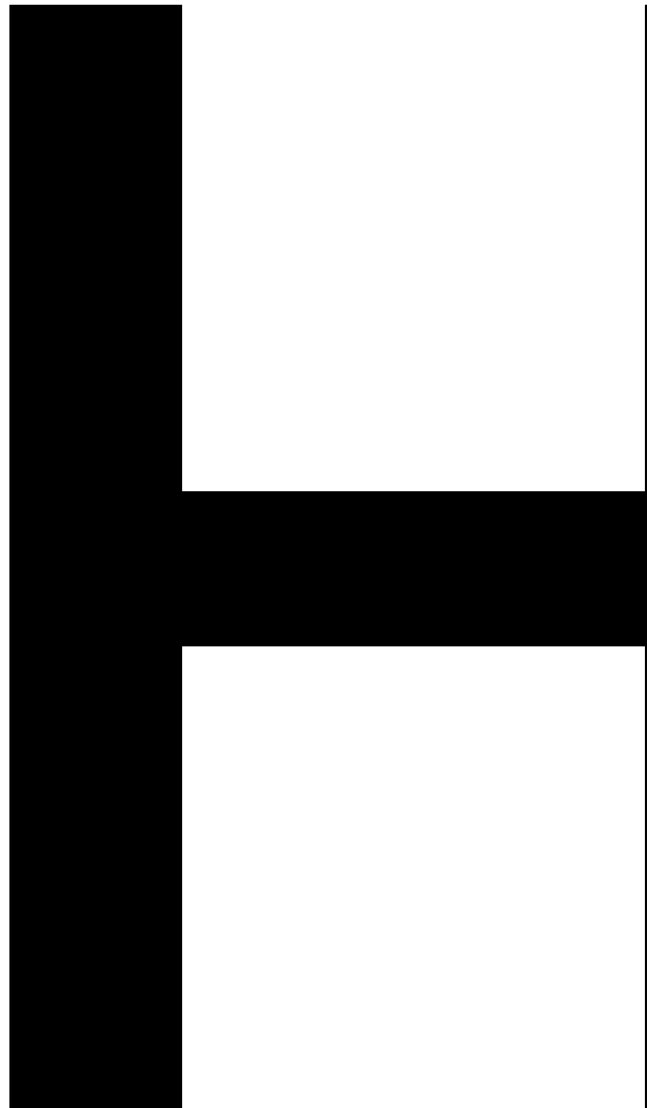


Holy Wars

Essay by Marc de Bruijn

MA Media Design

Piet Zwart Institute, 2006



The Right Thing or Worse is Better?!

01

The Internet has always been a place of vivid discussion. In the early days people exchanged ideas on mailing lists, local BBSs (Bulletin Board Systems) and later on Usenet. The coming of the World Wide Web ignited a rapid growth in user communities who are actively discussing many different topics. Quite often heated debates evolve into so called “holy wars”.

“holy wars: n.

*[from Usenet, but may predate it; common] n. flame wars over religious issues. The paper by Danny Cohen that popularized the terms big-endian and little-endian in connection with the LSB-first/MSB-first controversy was entitled *On Holy Wars and a Plea for Peace.*”*

[Eric S. Raymond, “Jargon File version 4.4.7”, <http://catb.org/jargon>, 2003]

First some background to the phenomenon. Multics (Multiplexed Information and Computing Service) was an operating system developed at MIT (Massachusetts Institute of Technology), with the help from General Electric and Bell Labs from 1965 until 2000. It was intended to be a commercial operating system for General Electric, but it only got to that status when Honeywell International took over the project. After Bell Labs and General Electric pulled out of the project, Ken Thompson – a former employee of Bell Labs – went on to create UNICS (Uniplexed Information and Computing System), together with Dennis Ritchie and a team of developers. UNICS eventually became UNIX and the basis of every GNU/Linux and BSD system. Multics has never been as popular as UNIX, despite the fact that the system was developed from the beginning to scale with the power of the hardware. Important in this case is the fact that Multics was initially developed by MIT and UNIX by people from Bell Labs in New Jersey and later on by the University of California in Berkeley (Berkeley Software Distribution, BSD). Those two facts feature prominently in the discourse by Richard P. Gabriel, “The Rise of ‘Worse is better’”, part of the essay “Lisp: Good News, Bad News, How to Win Big”. In the classic section of that article he analyses his work process as a Common Lisp and Common Lisp Object System developer (CLOS). Gabriel identifies two distinct design approaches in this essay, “New Jersey style” (aka “Worse is better”) and the “MIT approach” (or “the Right Thing”). Gabriel describes the key differences between the two approaches by focussing on simplicity, correctness, consistency and completeness. He argues that the MIT approach values simple design with the focus on a simple interface rather than simple implementation. Incorrectness and inconsistency are not permitted and equally important. The design must be complete, meaning that all the expected functionality must be covered, simplicity isn’t allowed to reduce completeness too much. According to Gabriel the “New Jersey” style is only slightly different, but different nevertheless. In this philosophy simplicity is the most important feature of the four, and one should give absolute priority to the simplicity of the implementation, rather than the simpleness of the interface. The design should be correct in every way, but a design is allowed to be simpler and less correct. Consistency isn’t the Holy Grail, one is allowed to be slightly inconsistent, but it’s advised to drop the part which creates implementation complexity altogether. Interface consistency is not a priority. Lastly, a design must cover as many

practical cases in order to be complete. But completeness must be sacrificed when implementation simplicity is threatened or when the developer wishes to give priority to the other qualities. Consistency can be dropped to achieve completeness.

Gabriel identifies C and (early) UNIX as being developed along the lines of the “Worse is better” doctrine and Lisp and Multics a product of “the Right Thing” idea. While Gabriel doesn’t agree with the “Worse is better” approach he does however see more survival value for it in comparison to the “MIT approach”. He argues that UNIX and C run on low-end hardware, because of their design policy, and sees in this the reason why UNIX and C ultimately dominated the world of computer technology, instead of Multics and Lisp.

“Half the computers that exist at any point are worse than median (smaller or slower). Unix and C work fine on them. The worse-is-better philosophy means that implementation simplicity has highest priority, which means Unix and C are easy to port on such machines. Therefore, one expects that if the 50% functionality Unix and C support is satisfactory, they will start to appear everywhere. And they have, haven’t they?”

Unix and C are the ultimate computer viruses.”

(Richard P. Gabriel, “The Rise of ‘Worse is Better’”, <http://www.jwz.org/doc/worse-is-better.html>, 1989)

Apart from backing up the theory of “Worse is better” being the fittest strategy to survive, both design approaches are interesting. Mainly because they still play a role in the various “holy wars” in the realm of the Free and Open Source software community, or they are the reason why those wars are fought in the first place.

Eight Megabytes And Constantly Swapping versus vi vi vi

02

For somebody who hasn't got any affinity with the GNU/Linux or Free and Open Source software community as a whole, the intensive debates on various topics might seem tedious and utterly nonsensical. While it might be hard to explain a lay what the significance is of a long dispute over the best desktop environment in existence, KDE versus GNOME, the disputes themselves are quite an interesting thing to look at, as they give us a nice overview of the philosophical and ideological contrasting ideas on usability, software design, etc. floating around in the FLOSS community.

One of the most famous "holy wars" is the rivalry between two text editors, vi(m) and Emacs. Richard Stallman wrote Emacs, with the help of Guy Steele, in 1975 and it became, albeit heavily changed in comparison to the original version, the first release of Stallman's Free Software Foundation (FSF) as GNU Emacs in 1984. New versions of Emacs are still released by the FSF, as well as forked projects like XEmacs. vi was created by Bill Joy for the BSD platform in 1976, Bram Moolenaar created Vi IMproved (vim) in 1991. vim is a direct descendant of the original vi but extended with more functionality and is still updated as of today. The key differences and supposed supremacy of one editor above the other is still the fuel of heavy debate between the Emacs and vi camp.

One of the main differences is that Emacs is traditionally more used as a GUI text editor (although it can be run without a GUI as well), vi doesn't come with a graphical interface and is commonly used as a strictly command line programme. The modal aspect of vi, the user has to switch between text editing and command mode, as opposed to Emacs where there is no difference between input or command mode. Furthermore, Emacs is fully customizable and extendible, whereas vi has a fixed set of features which are only updated at each new release.

More interesting is the interpretation of the UNIX philosophy in both programmes. Proponents claim that vi, as a monolithic programme, adheres to the UNIX guidelines – especially the rule "write programs that do one thing and do it well", as opposed to Emacs which is a multi-purpose tool. Advocates of Emacs however, claim that "their" programme complies with the UNIX guidelines, because of its customisability and extendibility, and vi does not because of its lack of customisability.

The war, as well as the arguments itself have become some sort of icons of net culture. An example of this is the pun on Emacs relying very much on modifier keys during the edit process. This has led to an alternative acronym of Emacs (instead of the original "Editor MACroS"), like "Escape Meta Alt Control Shift". Richard Stallman went as far as calling himself St IGNU-cius, saint of the Church of Emacs, mocking the constant controversy and the zealous defending of both software applications. The vi community responded by establishing the Cult of vi.

The text editor war isn't a unique example, there are many instances of such wars in the UNIX community, ranging for disputes on hardware, code indentation style and the ultimate operating system.

Warring in the FLOSS community is quite heavy, because of the generally intensive involvement of the warring factions (consisting of hackers, software developers, etc.) in the disputed material. Often the actual wars focus on marginal details backed up by personal experiences of the persons involved in the conflict, while in the background a meta discussion is taking place on what a great UNIX programme should be like. A nice example of this is the GNOME versus KDE debate. Both GNOME and KDE are desktop environments with its own philosophy on accessibility, extendibility and ease of use. GNOME focusses heavily on simplicity for the user, while KDE aims to offer optimal configurability. This has led to opponents calling KDE "too complex" and "too configurable" and GNOME overly simple and only suited for "stupid users" and. Those exact sentiments were expressed by Linus Torvalds in a seemingly unrelated discussion of the design of a printer dialog in GNOME on the GNOME mailing-list.

"This "users are idiots, and are confused by functionality" mentality of Gnome is a disease. If you think your users are idiots, only idiots will use it. I don't use Gnome, because in striving to be simple, it has long since reached the point where it simply doesn't do what I need it to do."

(Linus Torvalds, "[Desktop_architects] Printing dialog and GNOME", GNOME Mailing list, 2006)

All the other replies by Torvalds on the same mailing-list are not so much about the printer dialog in GNOME, but rather a rant about the fundamentals of GNOME. Soon the discussion, which started out as something quite small digresses into an argument on what a desktop environment should be. Torvalds is complaining about GNOME cutting functionality to meet with the goal of simplicity. He feels that the aimed simplicity is in fact an underestimation of the user and adds up to the inflexibility of GNOME.

"But hey, just continue to remove all that confusing functionality from Gnome. I don't care. I voted with my feet."

(Linus Torvalds, "[Desktop_architects] Printing dialog and GNOME", GNOME Mailing list, 2006)

His arguments are met with counter-arguments which are often diametrically opposed to Torvalds' points. GNOME advocates see the simplicity as one of the main assets of the desktop environment and the supposed inflexibility is compensated by the increased ease of use and uncluttered design. They argue that the alternative proposed by Torvalds, KDE, might be highly configurable but is unnecessarily complicated, user unfriendly and bloated. To their standards GNOME fulfills at least one of the main points of the UNIX philosophy as identified by Raymond; "Write simple parts connected by clean interfaces" and Gancarz "Avoid captive user interfaces". **(Eric S. Raymond, "The Art of UNIX Programming", <http://catb.org/~esr/writings/taoup/>, 2003. Mike Gancarz, "The UNIX Philosophy", <http://hebb.cis.uoguelph.ca/~dave/27320/new/unixphil.html>, 1994)** The developers of KDE have largely a different approach, while they also claim to adhere to the concept of optimal usability, they seem to prioritise Gancarz' "Allow the user to tailor the environment" more than GNOME does. These policies and the personal interpretation of those policies tends to get more important than the actual facts. This is clearly the case in the "Printer dialog and GNOME" discussion, where Torvalds states that he basically doesn't like GNOME and its underlying philosophy by taking on a marginal aspect of the whole programme. The eventual discussion isn't about the printer dialog anymore, but about Torvalds personal preference when it comes to a desktop environment and how GNOME's philosophy is flawed. After Torvalds ceased to push reasons for his personal dislike of GNOME and went on to discuss the actual problems at hand, the discussion became constructive again.

The same scenario can be seen in the vi versus emacs war where the main arguments have been reduced to sound-bites and discussion generally ends up in a war on (personal) philosophies.

The bickering on such a level can be quite valuable, but only regarding software which hasn't matured

as much as vi, KDE, emacs or GNOME. Those programmes each have their own distinct direction and ideology behind the development. In the case of GNOME and KDE it is useful to remember the two design philosophies as argued by Richard Gabriel. GNOME can be seen as the distribution following the “the Right Thing” approach, with its focus on a simple interface rather than a simple implementation. Something Linus Torvalds clearly doesn’t value highly. KDE can be seen as a programme developed along the lines of the “Worse is better” style. In this case, however, it cannot simply be argued that KDE, by adhering to “Worse is better” is superior to GNOME, because historical precedents (UNIX, C versus Multics, Lisp) dictate this. The critique of Jim Waldo on the original “Worse is better” article called “Worse is worse”, might explain why this is the case. Waldo states that the superiority of Lisp is an observation from the author’s (Gabriel) viewpoint and not necessarily the truth.

“But this doesn’t necessarily show that Lisp was in fact superior to C; it can just as easily be taken to show that the metrics that were cited in the article were not the ones that were taken to be most important by those choosing a programming language. The fact that C produced faster code, was easier to master, was easier to use in groups, and ran well on less expensive hardware were not considerations that Gabriel found important. But others did. On those metrics, the dominance of C as a programming language was an example of better is better, not worse is better.”

[Jim Waldo, “Worse is worse”, <http://www.artima.com/weblogs/viewpost.jsp?thread=24807>, 2003]

It all depends on what the priorities are of a developer or end-user, this also applies to the KDE versus GNOME debate.

Hello, I'm a Mac.

03

Apart from the FLOSS community on the Internet, there are also more recently established communities, which came into existence at the beginning of the 1990s. Mainly the Mac and Windows community are in this case interesting, because holy warring is a very common, if not the most important, practice in such communities. The reason or cause behind the holy wars is different from the FLOSS wars however.

Guy Kawasaki is generally considered to be the inventor of the practice of so-called "evangelism marketing". Kawasaki developed the theory surrounding evangelism marketing while he was working at Apple Computer in the 1980s. The job post "evangelist" already existed at Apple Computer, but Kawasaki interpreted the term in a completely different way. The traditional practice of "evangelising" was basically the company going out on the street and telling people how great the products were and that they should buy them. Kawasaki's idea was to create customers who were extremely brand loyal and vocally supportive of the company's products, while not being paid or otherwise supported by the company. In the case of Apple Computer it's evident that Kawasaki succeeded in his approach to create loyal customers. Kawasaki, with the help of others, created and exploited the image of the hip underdog who crushes the mainstream and uncreative colossus – IBM at that time. The famous "1984" commercial basically shows Apple represented by a girl with a sledgehammer, destroying "Big Brother". Apple Computer cultivated its image even further with its "Think Different" campaign using cultural icons like Bob Dylan, Albert Einstein and Alfred Hitchcock. Apart from the advertising work done by Apple, Steve Jobs is in this case a very important part of Apples appearance to the outside world. Jobs' charismatic performance at various product presentations, or "special events" as Apple persistently calls them, are quite unlike the presentations given at other companies. The way Jobs' is able to get an audience enthusiastic – or even more enthusiastic than they already were – about simple applications like a desktop manager, a backup programme or some new GUI elements is quite unique. With these events and appearance in the media Apple is carefully cultivating the image of the hip, small company established by Kawasaki in the 1980s. The actual size of the company isn't small of course, especially with Apple being the major player on the MP3 player market, but the advertising succeeds in somehow masking the fact that customers are still dealing with a large corporation trying to make a profit out of its products, instead of a benefactive saviour. The human face of the company, in combination with the aura of innovation and the focus on product design have laid the basis for a large customer community.

"He consistently and clearly identifies what Apple stands for and where it's headed, both to employees and customers. This is a hallmark feature of an organizational evangelist: Provide something of value for people to believe in. "Personal computers will be the digital hub of our new digital lifestyle," Jobs said at Macworld, firmly planting his company's stake in the ground."

[Ben McConnell & Jackie Huba, "The hitmaker", http://www.creatingcustomerevangelists.com/resources/evangelists/steve_jobs.asp]

Microsoft however, has been having difficulties with its image for years on end. The company is seen as a big monopolist having difficulties catching up with progress in the industry. Apart from this, Microsoft is associated by many as being greedy and allround evil. The fact that Microsoft is the largest player in the OS market plus their policies on selling Windows might not help this negative image, but there's no guarantee that Apple or Sun won't do the same when in the comfortable and powerful position of a monopolist. However, with Apples domination of the MP3 player market, the company's human face is still intact, despite being a large company (14,800 in 2005) with a total revenue of \$ 13.9 billion dollars in 2005. This is largely due to the careful image Apple has build up from the 1980s. The most important aspect of the evangelism by Apple Computer in the 1980s was selling a dream, rather than the product. The traditional way of advertising a product is more or less saying how great a product is, the method developed by Kawasaki was to sell an inspirational idea or a vision. The first Macintosh was marketed as a great technological innovation but just as much as a product which would change the world in one way or another. This message, with the help of the now iconic commercial, became the motivation of many people to actively propagate the use of the Macintosh and further fuel the image of the product being the promise of information technology in 1984.

While Microsoft seems to focus on different issues every five years (security being the theme nowadays), Apple hasn't changed its approach radically over the years. All the products are still marketed with message of the company's desire to change the world, including Mac OS X. That doesn't mean Microsoft doesn't have a message when it comes to marketing Windows, the keynotes by Bill Gates explaining the company's vision on the future of technology are evidence of this. The main difference is that they're not evangelising it.

The method of Apple is creating a strong "we" (the company and its customers) versus "them" (the competition) feeling. One of the best examples is the current "Get a Mac" advertising line. The advertisements basically feature two people, a young hipster and an older dull person, representing respectively a Mac and a PC talking to each other about various differences between the two products and the ultimate superiority of Apple products. This aggressive stance fuels the ongoing online debates of Mac users versus PC users, especially the ones running Microsoft Windows. Loyal Mac users feel at home in the "cocoon" created by Apple Computer, the brand itself has become the symbol of a message which is worth evangelising.

Kawasaki codified the importance of having a message in his book "Selling the Dream", which he wrote after his work at Apple. Ben McConnell and Jackie Huba identified this and five other important perquisites for successfully creating customer evangelists:

- "Continuously gather customer feedback*
- Freely share your knowledge*
- Create intelligent word-of-mouth networks*
- Encourage communities of customers to meet and share*
- Devise specialized, smaller offerings to get customers to bite*
- Focus on making the world, or your industry, better"*

(Ben McConnell & Jackie Huba, "Creating Customer Evangelists: How Loyal Customers Become a Volunteer Sales Force", <http://www.creatingcustomerevangelists.com>)

It seems that Microsoft is also trying to at least incorporate some of the above rules in the marketing strategies of a selection of their new products, curiously not its operating system. Zune, the media player designed by Microsoft, is an example of such a product. The Zune created a buzz on blogs

and news sites until its release in November 2006. Special sites dedicated to the Zune were established, some of them later to be revealed by Microsoft, a strategy known as “astroturfing” and used by Microsoft for a long time. By creating those sites Microsoft established the means to create a platform to spread the buzz and establish small customer communities. It’s the “intelligent word-of-mouth network” McConnell and Huba talk about in their book, but already created before the actual release of the product itself. The device marketed with the term “Welcome to the Social”, partly referring to feature where users were able to wirelessly transfer DRM wrapped songs from one Zune device to another. The visuals accompanying the marketing slogans show endless variations on cool, young people having great fun and playing music. But somehow, this approach is not paying off, people are skeptical of the large monopolist trying to be hip and market a product as a social tool. It’s basically Microsoft decade-long reputation of being (too) big backfiring at them.

Although Microsoft has a steady base of customers, they’re not as fiercely advocating the Microsoft brand as it is the case with Apple. Of course they have the advantage of producing the OS which many people think is the only proper way to interface with a computer, because of the apparent omnipresence of Microsoft in the commercial and domestic sector. The warring between Mac and PC users is not as useful as the wars in the FLOSS realm sometimes are. The arguments are often reformulations of claims stating in advertisements or personal experiences.

“Don’t even ask how I got here, but just know I’m a Mac user, and I’m proud of it! I suggest you actually look at the new macs out there. Apple computer makes the fastest personal computer available, running at 350Mhz. I know you Windows users. “That’s wrong, there’s a 500Mhz Pentium out there!” Ya right. Is it for sale yet? No!”

[Dan F***@worldnet.att.net, “WINDOWS SUCKS”, <http://cexx.org/snicker/onfire.htm>, 1997]**

The only ones who are really ultimately benefitting of this clash between Mac and PC users are the manufacturers of the products. The debate itself is one without an end, no matter how much evidence is presented there won’t be any resolution which platform, OS or product is the best. The FLOSS wars may have a positive impact on the development of a product, because everybody is free to improve and modify, whereas the warring between Mac and PC is basically an endless loop of the same arguments. In both cases personal observations are a major part of the argument. The hackneyed vi versus Emacs discussion for example, is in essence a discussion of personal preference, because both applications have matured and developed their own distinct philosophy. In that case, as with the KDE versus GNOME war, the discussion is only an arena for people to show their personal preference, rather than improving software. Of course discussion about certain features of a particular programme are valuable, even when compared to similar features in other programmes, but the shoot-out between different entities like Emacs or vim is ultimately not such a constructive discussion. In the same vein, the Mac versus PC debates is basically the comparison of two abstract entities, who’s content is mostly provided by years of marketing. Curiously the Mac versus PC battle is not so much about the hardware itself most of the time, but about the operating system. This is especially the case nowadays, with Apples switch to Intel processors and their aggressive marketing of Mac OS X against MS Windows. A good example of this is Bertrand Serlet, senior vice president of software engineering at Apple Computer, mocking the similarities between their OS X and Microsoft’s newest release Windows Vista. **[Bertrand Serlet, “WWDC Keynote 2006”, <http://www.engadget.com/2006/08/07/live-from-wwdc-2006-steve-jobs-keynote/>, 2006]**

A lot of the arguments expressed by Serlet are also floating around in the Mac versus PC debate.

Other memes include the mention of the system crashes (relatively common prior to Windows 2000) and referral to “Blue Screens of Death” appearing when Windows fatally crashes. The steep hardware requirements of Windows Vista and Windows XP at that time. Lastly, the accusations of Microsoft copying Apple Computer, which dates back from the mid 1980s with the release of Windows 1.0 using the same desktop metaphor as the Apple Macintosh.

Similarly arguments against “the Mac” include the limited gaming capabilities and availability of games as well as the alleged minimal variety of Mac software. A curious, but recursive argument is the one about Mac users being dumber than PC users, something apparently related to the easy-to-use philosophy of Apple.

“But many Mac users are oblivious to how files should be named, how folders can organize information, or how network volumes work. The problem occurs on both platforms, but I’ve never had a Windows user try to put a file on the web called ‘Mr. Smith’s web page .html.’”

(John Schinker, “Why I Hate Macs”, <http://staff.bbhcscd.org/schinkerj/archives/2006/04/19/why-i-hate-macs/>, 2006)

Who's helping who?

04

Apart from often being fruitless discussions, the warring in both Mac, PC and FLOS communities are interesting from a "cultural" point of view. In the case of FLOSS warring the difference in ideologies and development philosophy is interesting to filter out of the discussion. The Mac versus PC wars are interesting to study because of the huge impact of marketing on the discussion itself. In the case of the marketing fueled wars the clash between two design philosophies has been replaced by a distinct vision heavily influenced by advertising and strictly personal experiences.

But in the end, who's helping who? In the case of the FLOSS community it could be argued that warring can be an inspiration to improve or enhance a software application. However, if the discussion digresses to the level of flaunting ones personal preference, in other words seizing to discuss the actual content of an application, the debate eventually becomes useless. This is often the case with already matured and established programmes, like KDE, GNOME, vim or Emacs.

The same is happening in the wars fueled by "evangelism" marketing. Except for the fact that the first, constructive part of the discussion in FLOSS wars is virtually non existent in the marketed wars. The result of the those debates isn't improving or enhancing the product or applications discussed, but rather generating positive publicity for the manufacturer of the product.

Marc de Bruijn
MA Media Design, Piet Zwart Institute
Rotterdam, December 2006

References

05

- *"Jargon File version 4.4.7"*

Eric S. Raymond
<http://catb.org/jargon>
2003

- *"The Rise of 'Worse is Better'"*

Richard P. Gabriel
<http://www.jwz.org/doc/worse-is-better.html>
1989

- *"[Desktop_architects] Printing dialog and GNOME"*

Linus Torvalds
GNOME Mailing list
2006

- *"The Art of UNIX Programming"*

Eric S. Raymond
<http://catb.org/~esr/writings/taoup>
2003

- *"The UNIX Philosophy"*

Mike Gancarz
<http://hebb.cis.uoguelph.ca/~dave/27320/new/unixphil.html>
1994

- *"Worse is worse"*

Jim Waldo
<http://www.artima.com/weblogs/viewpost.jsp?thread=24807>
2003

- *"The hitmaker"*

Ben McConnell & Jackie Huba
http://www.creatingcustomerevangelists.com/resources/evangelists/steve_jobs.asp

- *"Creating Customer Evangelists: How Loyal Customers Become a Volunteer Sales Force"*

Ben McConnell & Jackie Huba
<http://www.creatingcustomerevangelists.com>

- *"WINDOWS SUCKS"*

Dan F*****←*****@worldnet.att.net
<http://cexx.org/snicker/onfire.htm>
1997

- *“WWDC Keynote 2006”*

Bertrand Serlet

<http://www.engadget.com/2006/08/07/live-from-wwdc-2006-steve-jobs-keynote/2006>

- *“Why I Hate Macs”*

John Schinker

<http://staff.bbhcsc.org/schinkerj/archives/2006/04/19/why-i-hate-macs/2006>

- *“A Simple Argument for the Mac”*

Peter Lewis

http://www.mindspring.com/~mcameron/simple_argument.html
1994

- *“Rules, Sins, Virtues, Gods and more of The Church of EMACS according to The Gospel of Prophet Antony”*

ramprasadb/The Free Software Foundation

<http://www.gnu.org/fun/jokes/gospel.html>
2006

- *“iPod vs Zune: Microsoft’s Slippery Astroturf”*

Roughly Drafted

<http://www.roughlydrafted.com/RD/Q4.06/D0BC712B-7DBA-46CA-AA44-19376E64FBA6.html>
2006

- *“The evangelist’s evangelist”*

Ben McConnell & Jackie Huba

http://www.creatingcustomerevangelists.com/resources/evangelists/guy_kawasaki.asp

- *“The Customer Evangelism Manifesto 1.02”*

Ben McConnell & Jackie Huba

<http://changethis.com/1.CustomerEvangelist>