

Worse is better

Worse is better, also called **New Jersey style**^[1], was conceived by Richard P. Gabriel in an essay "Worse is better" to describe the dynamics of software acceptance, but it has broader application. It is the idea that quality does not necessarily increase with functionality—that there is a point where less functionality ("worse") is a preferable option ("better") in terms of practicality and usability. Software that is limited, but simple to use, may be more appealing to the user and market than the reverse.

As to the oxymoronic title, Gabriel calls it a caricature, declaring the style bad in comparison with "The Right Thing". However he also states that "it has better survival characteristics than the-right-thing" development style and is superior to the "MIT Approach" with which he contrasted it in the original essay.^[2]

Contents

- Origin
- Description
- The MIT approach
- Effects
- See also
- References
- External links

Origin

Gabriel was a Lisp programmer when he formulated the concept in 1989, presenting it in his essay "Lisp: Good News, Bad News, How to Win Big". A section of the article, titled "The Rise of 'Worse is Better'", was widely disseminated beginning in 1991, after Jamie Zawinski found it in Gabriel's files at Lucid Inc. and e-mailed it to friends and colleagues.^[2]

Description

In *The Rise of Worse is Better*, Gabriel claimed that "Worse-is-Better" is a model of software design and implementation which has the following characteristics (in approximately descending order of importance):

Simplicity

The design must be simple, both in implementation and interface. It is more important for the implementation to be simple than the interface. **Simplicity is the most important consideration in a design.**

Correctness

The design should be correct in all observable aspects, but **It is slightly better to be simple than correct.**

Consistency

The design must not be overly inconsistent. Consistency can be sacrificed for simplicity in some cases, but it is better to drop those parts of the design that deal with less common circumstances than to introduce either complexity or inconsistency in the implementation.

Completeness

The design must cover as many important situations as is practical. All reasonably expected cases should be covered. **Completeness can be sacrificed in favor of any other quality.** In fact, completeness must be sacrificed whenever implementation simplicity is jeopardized. Consistency can be sacrificed to achieve completeness if simplicity is retained; especially worthless is consistency of interface.

Gabriel argued that early Unix and C, developed by Bell Labs, are examples of this design approach.

The MIT approach

Gabriel contrasted his philosophy with what he called the "MIT/Stanford style of design" or "MIT approach" (also known as "the Right Thing"), which he described as follows. Contrasts are in bold:

Simplicity

The design must be simple, both in implementation and interface. It is **more important for the interface to be simple** than the implementation.

Correctness

The design must be correct in all observable aspects. **Incorrectness is simply not allowed.**

Consistency

The design must be consistent. A design is allowed to be slightly less simple and less complete to avoid inconsistency. Consistency is as important as correctness.

Completeness

The design must cover as many important situations as is practical. All reasonably expected cases must be covered. **Simplicity is not allowed to overly reduce completeness.**

Effects

Gabriel argued that "Worse is better" produced more *successful* software than the MIT approach: As long as the initial program is basically good, it will take much less time and effort to implement initially and it will be easier to adapt to new situations. Porting software to new machines, for example, becomes far easier this way. Thus its use will spread rapidly, long before a program developed using the MIT approach has a chance to be developed and deployed (first-mover advantage). Once it has spread, there will be pressure to improve its functionality, but users have already been conditioned to accept "worse" rather than the "right thing". "Therefore, the worse-is-better software first will gain acceptance, second will condition its users to expect less, and third will be improved to a point that is almost the right thing. In concrete terms, even though Lisp compilers in 1987 were about as good as C compilers, there are many more compiler experts who want to make C compilers better than want to make Lisp compilers better."^[3]

Gabriel credits Jamie Zawinski for excerpting the worse-is-better sections of "Lisp: Good News, Bad News, How to Win Big" and e-mailing them to his friends at Carnegie Mellon University, who sent them to their friends at Bell Labs, "who sent them to their friends everywhere". He apparently connected these ideas to those of Richard Stallman and saw related ideas that are important in the design philosophy of Unix, and more generally in the open-source movement, both of which were central to the development of Linux.

Gabriel later answered his earlier essay with one titled *Worse Is Better Is Worse*^[4] under the pseudonym "Nickieben Bourbaki" (an allusion to Nicolas Bourbaki).

See also

- Perfect is the enemy of good
- Satisficing
- Unix philosophy

References

1. Worse is Better vs. Better is Better (<https://andrumyers.wordpress.com/2014/09/20/worse-is-better-vs-better-is-better>)
2. Worse Is Better (<http://dreamsongs.com/WorselsBetter.html>) (Richard P. Gabriel, personal recollection)
3. Lisp: Good News, Bad News, How to Win Big (<http://dreamsongs.com/WIB.html>), Richard P. Gabriel
4. Worse Is Better Is Worse (<http://dreamsongs.com/Files/worse-is-worse.pdf>) (PDF) (Richard P. Gabriel as "Nickieben Bourbaki")

External links

- Worse is Better (<http://dreamsongs.com/WorselsBetter.html>), Richard P. Gabriel
- Is Worse Really Better (<http://dreamsongs.com/Files/IsWorseReallyBetter.pdf>), Richard P. Gabriel

- [The Rise of "Worse is Better"](http://www.jwz.org/doc/worse-is-better.html) (<http://www.jwz.org/doc/worse-is-better.html>), Richard P. Gabriel
 - [Worse is Worse](http://www.artima.com/weblogs/viewpost.jsp?thread=24807) (<http://www.artima.com/weblogs/viewpost.jsp?thread=24807>), Jim Waldo (Sun Microsystems engineer)
 - Daniel Weinreb. "The "Worse is Better" idea and the future of Lisp" (<https://web.archive.org/web/20090611174852/http://danweinreb.org/blog/the-worse-is-better-idea-and-the-future-of-lisp>). Archived from the original (<http://danweinreb.org/blog/the-worse-is-better-idea-and-the-future-of-lisp>) on June 11, 2009.
 - Kreinin, Yossi (August 11, 2012). "What "Worse is Better vs The Right Thing" is really about" (<https://yosefk.com/blog/what-worse-is-better-vs-the-right-thing-is-really-about.html>). *Proper Fixation*. Retrieved 2018-11-24.
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Worse_is_better&oldid=873677787"

This page was last edited on 14 December 2018, at 12:33 (UTC).

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.