WIKIPEDIA

Software bloat

Software bloat is a process whereby successive versions of a computer program become perceptibly slower, use more memory, disk space or processing power, or have higher hardware requirements than the previous version— while making only dubious user-perceptible improvements or suffering from feature creep. The term is not applied consistently; it is often used as a pejorative by end users (bloatware) to describe undesired user interface changes even if those changes had little or no effect on the hardware requirements. In long-lived software, perceived bloat can occur from the software servicing a large, diverse marketplace with many differing requirements. Most end users will feel they only need some limited subset of the available functions, and will regard the others as unnecessary bloat, even if end users with different requirements require those functions.

Actual (measurable) bloat can occur due to de-emphasising software efficiency in favour of other concerns like developer productivity, or possibly through the introduction of new layers of abstraction like a virtual machine or other scripting engine for the purposes of convenience when developer constraints are reduced. The perception of improved developer productivity, in the case of practising development within virtual machine environments, comes from the developers no longer taking resource constraints and usage into consideration during design and development; this allows the product to be completed faster but it results in increases to the end user's hardware requirements to compensate.

The term "bloatware" is also used to describe unwanted preinstalled software or bundled programs.[1]

Contents

Causes

Software inefficiency

Software developers involved in the industry during the 1970s had severe limitations on disk space and memory. Every byte and clock cycle was taken into account, and much work went into fitting the programs into available resources. Achieving this efficiency was one of the highest values of computer programmers, and the best programs were often called "elegant", a term used by mathematicians to describe a proof which is tidy, parsimonious and powerful.

By the 21st century, the situation had reversed. Resources were perceived as cheap, and rapidity of coding and headline features for marketing seen as priorities.[2] In part, this is because technological advances have since increased processing capacity and storage density by orders of magnitude, while reducing the relative costs by similar orders of magnitude (see Moore's law). Additionally, the spread of computers through all levels of business and home life has produced a software industry many times larger than it was in the 1970s. Programs are now usually churned out by teams, directed by committees in software development studios (also known as software houses or software factories) where each programmer works on only a part of the whole, on one or more subroutines.

Finally, software development tools and approaches often result in changes throughout a program to accommodate each feature, leading to a large-scale inclusion of code which affects the main operation of the software, and is required in order to support functions that themselves may be only rarely used. In particular, the advances in resources available have led to tools which allow easier development of code, again with less priority given to end efficiency.

Another cause of bloat is independently competing standards and products, which can create a demand for integration. There are now more operating systems, browsers, protocols, and storage formats than there were before, causing bloat in programs due to interoperability issues. For example, a program that once could only save in text format is now demanded to save in HTML, XML, XLS, CSV, PDF, DOC, and other formats.

Niklaus Wirth has summed up the situation in Wirth's law, which states that software speed is decreasing more quickly than hardware speed is increasing.

In his 2001 essay *Strategy Letter IV: Bloatware and the 80/20 Myth*,[3] Joel Spolsky argues that while 80% of the users only use 20% of the features (a variant on the Pareto principle), each one uses different features. Thus, "lite" software editions turn out to be useless for most, as they miss the one or two special features that are present in the "bloated" version. Spolsky sums the article with a quote by Jamie Zawinski referring to the Mozilla Application Suite (which later became SeaMonkey):

> "Convenient though it would be if it were true, Mozilla is not big because it's full of useless crap. Mozilla is big because your needs are big. Your needs are big because the Internet is big. There are lots of small, lean web browsers out there that, incidentally, do almost nothing useful. [...] But being a shining jewel of perfection was not a goal when we wrote Mozilla."[4]

Software bloat may also be a symptom of the second-system effect, described by Fred Brooks in *The Mythical Man-Month*.

Bloatware

The term "bloatware" may be applied to software that has become bloated through inefficiency or accretion of features as outlined above.[3] The term also commonly refers to software preinstalled on a device, usually by the hardware manufacturer, that is mostly unwanted by the purchaser.

The term may also be applied to the accumulation of unwanted and unused software elements that remain after partial and incomplete uninstallation. These elements may include whole programs, libraries, associated configuration information, or other data. Performance may deteriorate overall as a result of such leavings, as the unwanted software or software components can occupy memory, waste processing time, add disk I/O, consume storage and cause delays at system startup and shutdown. In the worst cases, the leftover software may interfere with the correct operation of wanted software.[5] Bloatware can be easily removed if the user has root access on his or her smartphone, though the rooting process has its own advantages and disadvantages,[6] such as voiding the manufacturer's warranty, and that certain software refuses to run on rooted devices.

Examples

Comparison of Microsoft Windows minimum hardware requirements (for x86 versions)

Windows version	Processor	Memory	Hard disk
Windows 95[7]	25 MHz	4 MB	~50 MB
Windows 98[8]	66 MHz	16 MB	~200 MB
Windows 2000[9]	133 MHz	32 MB	650 MB
Windows XP[10] (2001)	233 MHz	64 MB	1.5 GB
Windows Vista[11] (2007)	800 MHz	512 MB	15 GB
Windows 7[12] (2009)	1 GHz	1 GB	16 GB
Windows 8[13] (2012)	1 GHz	1 GB	16 GB
Windows 10[14] (2015)	1 GHz	2 GB	16 GB

Apple's iTunes has been accused of being bloated by efforts to turn it from a simple media player to an e-commerce and advertising platform,[15][16] with former *PC World* editor Ed Bott accusing the company of hypocrisy in its advertising attacks on Windows for similar practices.[17] In 2019, Apple announced the impending closure of the program, a move described by a commentator from *The Guardian* as being "long overdue", stating that the program had "become baroquely bloated, a striking anomaly for a company that prides itself on elegant and functional design."[18]

Microsoft Windows has also been criticized as being bloated – with reference to Windows Vista and discussing the new, greatly slimmed down Windows 7 core components, Microsoft engineer Eric Traut commented that "This is the core of Windows 7. This is a collection of components that we've taken out. A lot of people think of Windows as this really large, bloated operating system, and that may be a fair characterization, I have to admit. It is large. It contains a lot of stuff in it. But at its core, the kernel and the components that make up the very core of the operating system actually is pretty streamlined."[19][20] Ed Bott also expressed skepticism, noting that nearly every operating system that Microsoft has ever sold has been criticized as 'bloated' on first release, even those now regarded as the exact opposite, such as MS-DOS.[21] Quoting Paul Thurrott, Bott agreed that the bloat stems from numerous enterprise-level features included in the operating system that were largely irrelevant to the average home user.

CD- and DVD-burning applications such as Nero Burning ROM have become criticized for being bloated.[22] Superfluous features not specifically tailored to the end user are sometimes installed by default through express setups.

A number of technology blogs have also covered the issue of increased bloatware on cell phones. However, they refer to a different issue, specifically that of wireless carriers loading phones with software that, in many cases, cannot be easily, if at all, deleted. This has been most frequently cited with respect to Android devices, although this phenomenon exists on phones running many other operating systems.[23][24]

Some of the most popular current messaging apps, which were previously only focused on instant messaging, has been criticized for being bloated due to feature creep.[25][26][27][28] WeChat added the additional features such as games, subscription service, WeChat Pay e-wallet,[25] news aggregator, e-commerce hub, e-government[26] feature, cinema booking system, restaurant finder and ridesharing company,[28] which has increased the size of an app from 2 MB in 2011 to 58 MB in 2018. Facebook Messenger, which has been separated from Facebook app, is similarly criticized for adding additional features such as games, bots and features which copied from Snapchat such as Messenger Day (Stories), face filters, camera with ability to edit photos, draw doodle and add emojis and stickers.[29][30] In January 2018, the head of Facebook Messaging, David A. Marcus, admitted that the app itself are extremely bloated and promised to redesign the whole app to remove unnecessary features and streamline the app.[27] The redesigned and streamlined Facebook Messenger app are announced in October 2018, which its features are reduced to only messaging, stories, discover tab and camera.[31]

Alternatives

Some applications, such as GIMP, and software with additional functionality from plug-ins, use extensions or add-ons which are downloaded separately from the main application. These can be created by either the software developer or by third party developers. Plug-ins, extensions, and add-ons add extra functionality which might have otherwise been packaged in the main program.

Allowing these plug-ins, extensions, and/or add-ons reduces the space used on any one machine, because even though the application, the "plug-in interface", and all the plug-ins combined are larger than the same functionality compiled into one monolithic application, it allows each user to install only the particular add-on features they require, rather than forcing every user to install a much larger monolithic application that includes all of the available features. This results in a "stripped-down" or "out-of-the-box" application that is delivered in a compact package yet is ready for users to add any missing functionality.

Open source software may use a similar technique using preprocessor directives to include features at compile time selectively. This is easier to implement and more secure than a plugin system, but has the disadvantage that a user who wants a specific set of features must compile the program from source.

Sometimes software becomes bloated because of "creeping featurism"[32] (Zawinski's law of software envelopment). One way to reduce that kind of bloat is described by the Unix philosophy of "writing programs that do one thing and do it well," and breaking what would be a single, complicated piece of software into numerous simpler components which can be chained together using pipes, shell scripts, or other forms of interapplication communication.

See also

- Code bloat
- Feature creep
- Lightweight software
- GNU C library
- Minimalism (computing)
- Software rot
- Software suite
- Technical Debt
- Wirth's law
- Zawinski's law of software envelopment

References

1. whatisbloatware.com
2. Raymond, Eric S. (17 September 2003). *The Art of Unix Programming* (http://www.catb.org/~esr/writings/taoup/html/why_not_c.html) (1 ed.). Addison-Wesley Professional. ISBN 0131429019. Retrieved 16 June 2007. (See also: The Art of Unix Programming)
3. Spolsky, Joel (2001). "Strategy Letter IV: Bloatware and the 80/20 Myth - Joel on Software" (http://www.joelonsoftware.com/articles/fog0000000020.html). Retrieved 2 October 2006.
4. Zawinski, Jamie (1998). "easter eggs" (https://www.jwz.org/doc/easter-eggs.html). Retrieved 9 January 2008.
5. Santo Domingo, Joel (13 February 2012). "How to Remove Bloatware" (https://www.pcmag.com/article2/0,2817,2400136,00.asp). *PC Magazine*.

6. Mukesh, Bhavesh (4 March 2016). "Advantages & Disadvantages of Rooting Android Smartphones" (http://www.theandroidportal.com/advantages-disadvantages-rooting-android-smartphones/). *The Android Portal*.
7. "Microsoft KB: Windows 95 Installation Requirements" (http://support.microsoft.com/kb/138349/). Retrieved 22 July 2007.
8. "Microsoft KB: Minimum Hardware Requirements for a Windows 98 Installation" (http://support.microsoft.com/kb/182751/). Retrieved 22 July 2007.
9. "Windows 2000 Server Getting Started: Chapter 3 - Planning Your Windows 2000 Server Installation" (http://www.microsoft.com/technet/prodtechnol/windows2000serv/proddocs/srvgs/sgsch03.mspx#EMD). Retrieved 29 August 2007.
10. "Microsoft KB: System requirements for Windows XP operating systems" (http://support.microsoft.com/kb/314865/en-us). Retrieved 22 July 2007.
11. "Microsoft KB: System requirements for Windows Vista" (http://support.microsoft.com/kb/919183/). Retrieved 22 July 2007.
12. "Microsoft: System requirements for Windows 7" (http://windows.microsoft.com/en-us/windows7/products/system-requirements). Retrieved 9 October 2009.
13. "Windows 8 Release Preview ISO images" (http://windows.microsoft.com/en-US/windows-8/download). Retrieved 12 August 2012. "Windows 8 Release Preview works great on the same hardware that powers Windows 7"
14. "7 things to remember if you're trying out the Windows 10 preview" (http://www.computerworld.com/article/2690817/7-things-to-remember-if-youre-trying-out-the-windows-10-preview.html). www.computerworld.com. Retrieved 29 March 2015.
15. Streza, Steve (7 March 2007). "What happened to iTunes?" (http://stevestreza.com/2007/03/07/what-happened-to-itunes/).
16. Buchanan, Matt (12 October 2009). "iTunes 9 Will Be a Bloated Social Monster" (https://gizmodo.com/5335754/itunes-9-will-be-a-bloated-social-monster). *Gizmodo*. Retrieved 14 January 2010.
17. Bott, Ed (3 October 2008). "Slimming down the bloated iTunes installer" (https://www.zdnet.com/article/slimming-down-the-bloated-itunes-installer/). *ZDNet*. Retrieved 14 January 2010.
18. Naughton, John (9 March 2019). "Farewell then, iTunes, and thanks for saving the music industry from itself" (https://www.theguardian.com/commentisfree/2019/jun/09/farewell-itunes-thanks-for-saving-music-industry-from-itself). *The Guardian*. Retrieved 10 March 2019.
19. McDougall, Paul (28 January 2008). "Software Tool Strips Windows Vista To Bare Bones" (http://www.informationweek.com/news/showArticle.jhtml?articleID=205920302). *informationweek.com*.
20. Bott, Ed (31 March 2008). "Is MinWin really the new Windows 7 kernel?" (https://www.zdnet.com/blog/bott/is-minwin-really-the-new-windows-7-kernel/418).
21. Bott, Ed (13 March 2006). "Windows bloat? It's always been that way" (https://www.zdnet.com/article/windows-bloat-its-always-been-that-way/).
22. Cassia, Fernando (27 February 2007). " 'Nero Lite' and 'Nero Micro': smaller sometimes is better" (https://web.archive.org/web/20070302073052/http://www.theinquirer.net/default.aspx?article=37873). The Inquirer. Archived from the original (http://www.theinquirer.net/default.aspx?article=37873) on 2 March 2007. Retrieved 7 March 2007.
23. Gartenberg, Michael (12 September 2010). "Entelligence: Will carriers destroy the Android vision?" (https://www.engadget.com/2010/09/12/entelligence-will-carriers-destroy-the-android-vision/). Engadget. Retrieved 21 September 2010.
24. Milian, Mark (15 June 2010). " 'Junkware' comes standard on Verizon, T-Mobile smart phones" (http://latimesblogs.latimes.com/technology/2010/07/android-junkware.html). *Los Angeles Times*. Retrieved 21 September 2010.
25. Bischoff, Paul (12 February 2014). "What WeChat needs to learn from Facebook" (https://www.techinasia.com/wechat-learn-facebook). *Tech in Asia*. Retrieved 16 February 2020.

26. Chen, Laurie (22 July 2018). "Why China's tech-savvy millennials are quitting WeChat" (https://www.scmp.com/news/china/society/article/2156297/how-growing-privacy-fears-china-are-driving-wechat-users-away). *South China Morning Post*. Hong Kong. Retrieved 16 February 2020.
27. Constine, Josh (16 January 2018). "Bloated Facebook Messenger promises to simplify in 2018" (https://techcrunch.com/2018/01/16/messengrrrr/). *TechCrunch*. Retrieved 16 February 2020.
28. "Facebook opens up Messenger to third-party apps" (https://www.bbc.com/news/technology-32036322). *BBC*. 25 March 2015. Retrieved 16 February 2020.
29. Brown, Aaron (7 February 2017). "Facebook almost ruined WhatsApp, and now it has ruined Messenger" (https://www.express.co.uk/life-style/science-technology/763053/Facebook-Login-Messenger-vs-WhatsApp). *Daily Express*. Retrieved 16 February 2020.
30. Welch, Chris (18 May 2017). "How to replace Facebook Messenger with a much simpler, lighter version" (https://www.theverge.com/2017/5/18/15659414/facebook-messenger-lite-how-to-install). *The Verge*. Retrieved 16 February 2020.
31. Price, Rob (23 October 2018). "Facebook is giving Messenger an overhaul to try and cut down on bloat — and adding a dark mode" (https://www.businessinsider.my/facebook-redesigns-messenger-dark-mode-2018-10/?r=US&IR=T). *Business Insider*. Retrieved 16 February 2020.
32. Adams, Ernest (1 May 2007). "The Designer's Notebook: "creeping featurism produces a bloated, complicated mess" (http://www.gamasutra.com/features/20070501/adams_01.shtml).

Retrieved from "https://en.wikipedia.org/w/index.php?title=Software_bloat&oldid=982910994"

This page was last edited on 11 October 2020, at 03:32 (UTC).