The Life Cycle of Open Source Software

Jeffrey Elkner

University of the People

BUS 5112: Marketing Management

Dr. Soha Ragab

May 3, 2020

The Life Cycle of Open Source Software

The product life cycle (PLC) is "a process that traces the stages of a product's acceptance, from its introduction (birth) to its decline (death)" (Thompson VSE, 2013, Timestamp 0:20). The PLC consists of four stages: introduction, growth, maturity, and decline (Thompson VSE, 2013, Timestamp 0:42). The textbook, Principles of Marketing, defines the PLC as "the stages the product goes through after development (The University of Minnesota, 2010, p. 197), while the nibusinessinfo.co.uk website includes development as the first stage of the PLC and groups growth and maturity together, thus still keeping the number of stages to four (Product Life Cycle, 2019). The question as to whether the design process is preliminary to the product life cycle or a constituent part of it becomes more important when the product under development is a computer program. In software development design and manufacturing occur continuously in an iterative process which can not be separated neatly into before and after. When the product is open source software the situation becomes even more intertwined, since the number of designers can greatly increase and the possibility of product forks exists. This paper will take a brief look into the life cycle of open source software.

In episode 8 of his Clean Code video series, titled, "Foundations of the SOLID Principles", Robert C. Martin discusses how the product life cycle of computer software is greatly impacted by the inverted cost structure of software design (2012, Timestamp 16:34). In most examples of product development, the cost of designing the product is much less expensive than the cost of manufacturing it, and the cost of correcting flaws in the design after manufacturing begins can be huge. For this reason a great deal of effort is put into the design process before the product is manufactured (Martin, 2012, Timestamp 11:14). With software the

reverse is true. The cost of designing software, which is done by programmers, is expensive compared to the cost of building it, which is done automatically by computer programs called compilers and interpreters for a tiny fraction of the cost of design. Fixing flaws after the fact is also inexpensive for the same reason, since only the design needs to be changed, not the building process (Martin, 2012, Timestamp 15:42). Martin describes a thought experiment to illustrate this point. How would you go about designing a house, he asks, if you could build a house for a hundred dollars, and it took an hour? You would not hire an expensive architect to create the final design, but would instead sketch a few rooms on the back of a napkin, build it, then move in and see how you liked it. As you lived in your house you would take notes on things you liked and did not like, sketch new designs, and periodically make changes to your house, since you could do this in an hour for a hundred dollars (Martin, 2012, Timestamp 16:47). For as long as you lived in the house this iterative process would never stop, though the rate of change would likely slow over time (Martin, 2012, Timestamp 18:35).

The logic of this cost inversion impacts the process of designing and building computer programs, as well as the resulting product life cycle. The PLC is further impacted when the software is free or open source software. The Free Software Foundation defines free software as "software that respects users' freedom and community" and states that this roughly means that "users have the freedom to run, copy, distribute, study, change and improve the software" (Free Software Foundation, 2019). Since free software is essentially public property, and since a portion of its consumers can and often do become its producers in turn, free software is subject to the network effect, whereby the more people use a particular computer program the more value it has for all its users (Wikipedia contributors, 2020, April 28). Since users of free software have

the freedom to share it with others, and since the cost of sharing it is essentially zero, the seemingly paradoxical situation emerges where the more free software is given away, the more it is worth. Value in this context is use value, not exchange value. This has a direct impact on the development process and life cycle of open source software.

In their white paper written for the Linux Foundation titled "Understanding the Open Source Development Model", authors Ibrahim Haddad and Brian Warner contrast the proprietary software development model, which generally consists of discrete periods of activity cascading towards a product's release, with the open source development model, which features a "more fluid development process characterized by increased intra-team collaboration, continuous integration and testing, and greater end-user involvement (2011, p. 1). The life cycle of free software is a continuous process of communication and feedback among the contributors, subsystem maintainers, principle maintainer, and the marketplace and follows the adage "release early and often" (Haddad & Warner, 2011, p. 2). It is characterized by an interwoven, iterative development cycle with constant peer review (Haddad & Warner, 2011, p. 6). In concluding their white paper, the authors state that "open source development model has proved to be very successful, with hundreds of success stories. This development model has special characteristics that allow faster development by broadly distributed teams, continual and thorough testing, faster innovation, multiple layers of peer review, and total openness and transparency throughout the project (Haddad & Warner, 2011, p. 7).

References

Free Software Foundation (2019, July 30). What is free software? Retrieved May 3, 2020 from https://www.gnu.org/philosophy/free-sw.html

Haddad, I., & Warner, B. (2011, November). Understanding the open source development model. *The Linux Foundation*. Retrieved May 3, 2020 from https://www.ibrahimatlinux.com/uploads/6/3/9/7/6397792/00.pdf

Martin, R. C. (2012, January). Foundations of the SOLID principles [video]. Available from https://cleancoders.com/episode/clean-code-episode-8

Product Life Cycle (2019, May 31). *Product life cycle - key stages*. Retrieved May 2, 2020 from https://www.nibusinessinfo.co.uk/content/product-life-cycle-key-stages

The University of Minnesota. (2010). *Principles of marketing.* Licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0. Retrieved April 9, 2020 from https://open.lib.umn.edu/principlesmarketing/front-matter/publisher-information/

Thompson VSE (2013, December 1). Product life cycles development, design and beyond free principles of marketing video [Video]. *YouTube*. Retrieved May 2, 2020 from https://www.youtube.com/watch?v=3s3aBhH0fAE&feature=youtu.be

Wikipedia contributors. (2020, April 28). Network effect. In *Wikipedia, The Free Encyclopedia*. Retrieved May 3, 2020 from https://en.wikipedia.org/w/index.php?title=Network_effect&oldid=953670346