Joseph A. Kaufman

Professor Henderson

CSCI 325

06 November 2021

Ethics of Rushing Code to Market

The realm of technology moves fast and is always changing. Software developers are expected to produce quality products that satisfy customers' needs while still making a profit. This is where a core ethical dilemma begins. Professionals are often forced to decide between selling unpolished buggy programs for profit or waiting to release polished code in sacrifice of profits. Where should developers draw the line of valuing customers and making a profit? It may not seem like a big issue starting off. Just pick one or the other and accept the consequences one might say. However, there are benefits and drawbacks to either one.

A benefit of going to market quickly is attaining a larger market share which increases profitability (Buzzell). This is accomplished by releasing a cutting-edge product before competition does. This increases the revenue and appeases the business side of the company. In addition to profit, the company will have vital customer feedback preventing them from being one of the 6 of 7 products that flop (Vanalli and Cziulik). This gives the company insight into whether they should scrap the product before major development costs are realized. Failure to release products in a timely fashion will put firms at a disadvantage. Speed is of the essence and so is profitability. If producing products is no longer profitable, innovation will dwindle, and firms will exit the market.

Releasing products too quickly has its negatives. Releasing a product too quickly may fail to live up to expectations. If it is not as promoted it can result in legal action. Rapidly produced

programs also are notorious for poor code practices and bugs. In 2018 alone, the total cost of poor code quality in the US was $2.84 Trillion (Krasner). In addition to code quality, extensive testing can be overlooked. This is especially dangerous if the program is responsible for the safety of others. For example, if the technology enables a self-driving car, the expectation for the technology is meeting robust testing requirements prior to sale. If a firm fails to implement proper procedures, the product could result in the death of people and damage to property. Therefore, selling a product hastily may be seen as misleading and negligent toward others. (Turley)

A good professional is aware of ethical situations and plans for them in advance. If a programmer is shipping buggy code, the programmer takes a few things into consideration. The first thing they need to do is account for accurate promotion of the product. Incomplete products should be labeled as a beta version and be discounted in price. Beta versions can create profitable code in the long run by allowing for feedback at early stages in development. This enables the company to cover costs and avoid deceptive practices. In addition, a programmer should be knowledgeable about proper practices and testing. Proper practices are vital to the long-term sustainability of code and the prevention of bugs. This makes the project run smoother and cost less to maintain (Krasner).

The other aspect of the software development dilemma is safety. This involves health care, energy, and other vital infrastructure. Preventing unnecessary outages or malfunctions, programmers should be especially cautious in circulating new software in these industries. The creation of architecture with strict safety protocols must happen prior to coding (Leveson). This enables the implementation of safety measures/warnings once coding has begun. All these procedures increase development costs which can make producing normal software infeasible.

However, patents can negate the extra costs by decreasing competition. This means they will have a larger market share than normal. Ultimately programmers should never rush products involving safety.

As an aspiring developer I feel prepared for most ethical situations. I read God's Word which is the best way to be prepared. "All Scripture is breathed out by God and profitable for teaching, for reproof, for correction, and for training in righteousness (*English Standard Version*, 2016)." By reading scripture, it will fine tune my conscience to know what to do in an ethical dilemma. I believe it is unethical to sell poor quality code that has not been tested properly. My code quality is not where it needs to be. However, a byproduct of being a student means it increases greatly each semester. Another area of possible improvement is learning to test code. I can improve by taking online certificate courses to be better prepared for the workforce. Ultimately by doing my job right, it will prevent the unintentional harm of others.

Programmers are not totally in the dark when it comes to ethics. Ethical societies such as the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE) seek to create clarity in professional conduct. However, ethical authority will always lie in God's unchanging Word. One can compare the Bible with section 2.3 of The ACM Code of Ethics and Professional Conduct. It states that professionals should "know and respect existing rules pertaining to professional work". This aligns with the Biblical idea that men should respect authority because authority is instituted by God (*English Standard* Version, 2016, 2 Peter 2:13-17). Another example of Biblical truths in ethics is in IEEE's Code of Ethics. In section 1.5 it states that programmers are "to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others." Proverbs talks about honesty

when it states, “the righteous hate what is false, but the wicked make themselves a stench and bring shame on themselves (*English Standard Version, 2016,* 13:5).” This IEEE ethical concept parallels the Bible because God loves truth and falsehood is a result of sin. Although not used in Biblical context, both professional standards help make professionals successful and produce good work.

Works Cited

"ACM Code of Ethics and Professional Conduct." *ACM.org*, Association for Computing Machinery, https://www.acm.org/code-of-ethics.

Buzzell, Robert D., et al. *Market Share-a Key to Profitability*. https://jvminc.com/Clients/JVP/HBR_Market_Share.pdf.

ESV: Study Bible: English Standard Version. Crossway Bibles, 2016.

"IEEE Code of Ethics." *IEEE*, Institute of Electrical and Electronics Engineers, https://www.ieee.org/about/corporate/governance/p7-8.html.

Krasner, Herb. *The Cost of Poor Quality Software in the US: A 2018 Report*. Consortium for IT Software Quality (CISQ), https://www.it-cisq.org/the-cost-of-poor-quality-software-in-the-us-a-2018-report/The-Cost-of-Poor-Quality-Software-in-the-US-2018-Report.pdf?source=post_page----------.

Leveson, Nancy G. "The Therac-25: 30 Years Later." *Computer*, vol. 50, no. 11, 2017, pp. 8–11., https://doi.org/10.1109/mc.2017.4041349.

Turley, Jim. "Can You Get Sued for Bad Code?" *EEJournal*, 29 Apr. 2017, https://www.eejournal.com/article/20161116-liability/.

Vanalli, S., and C. Cziulik. "Seven Steps to the Voice of the Customer."