# Information Presentation Response

1. I did not know of general software certifications like IEEE and ACM. I have always known them as general good practices, not as something you can be certified in. This is interesting because if we can create certifications for small industries of software, maybe there is a chance this can spread around and create more certifications in the future.
2. Further discussion on software engineering certifications would be very interesting, it’s a topic frequency talked about in traditional engineering, but some people studying computer science don’t even know that “engineer” is a professional title that needs certification to hold.

# Ethical Advice

To summarize, both Sam and Cheryl have discontent with providing their product to companies with practices they do not agree with and seem willing to sabotage the product (actively participating or not) which they are selling to their undesired customer. In addition, Sandra, the president, does not want the company to go bankrupt but has concerns about how her employees acted about selling their product too. Because this could affect their future sales as there are a fair number of people opposed to the practices of their potential customers, and it could seem like their company supports these practices by selling their product to them.

As the president of the company, Sandra has a professional responsibility to make decisions which are in the best interest of the company, but her conflict lies in the fact that her coworkers do not wish to sell their product because of their personal moral code. Some research or analysis could be done to see what effects “supporting” these practices could have on their future potential customers, or perhaps try to keep the information of their customers private so as to not associate the beliefs of the company’s customers to the company values themselves, but either way because of the nature of the Canadian (and more generally the North American) market, a business must prioritize profits less they lose their purpose because of competition.[1] The potential harm is obviously losing these customers and further pushing their company toward bankruptcy, and the potential benefits are potentially saving the company from bankruptcy. While both homoeopathy and abortion are legal practices, she should take into account the ethical portion of the situation and balance the two according to research and abide by any responsibilities and duties the company has.

Cheryl’s case has issues of letting their personal beliefs dictate if they want to intervene in the company's decisions or not. While there should be a balance between upholding your professional responsibilities and keeping your choices ethical to your personal morals [2], the choice to ignore your professional duties which these two made has potentially large negatives to the survival of the company. In accordance with professional ethics, it would perhaps be better for them to properly discuss their concerns with the company and its stakeholders rather than outright refuse to work because of their personal beliefs, and ultimately leave the decision to the company.

Furthermore, Sam’s idea of sabotaging the product by introducing bugs to the software is unethical to a higher degree, both professionally and generally, as you completely undermine any integrity the company has and your personal beliefs and potentially harm many people who are connected to the abortion clinic and homoeopath clinic. Moreover, this action is very likely illegal as well, which will further damage the already declining company. The act of sabotaging a product delivered to a company which works in a sector against your personal belief is simply not justifiable, not only does this potentially harm many unrelated people to the situation as the product being sold manages medical documents, but also benefits essentially no one in a utilitarian, Kantian, or virtuous way, which is in direct contradiction to IEEE/ACM Software Engineering ethics.[4]

# Report on a Software Disaster

The Therec-25 is a computerized radiation therapy machine known for causing multiple accidents involving radiation overdose, resulting in numerous injuries and deaths. The general use of the Therec-25 was to use high-energy beams (radiation) to target tumours while trying to minimize the impact on the surrounding tissue. Developed mainly by AECL (Atomic Energy of Canada Limited), the creation of the Therec-25 was from two earlier models called Therac-6 and Therac-20, but unlike these earlier models, the Therec-25 only used computer controls rather than being based on machines which already had clinical use. The software for the Therac-25 was developed using PDP 11 assembly and reused a lot of existing code from the earlier models. This software was responsible for keeping track of the status of the machine, preparing for any treatments the operator sets, and executing any commands the operator gives, such as turning on/off the radiation beam or printing out its display. There were six total accidents during the period between June 1985 and January 1987 and AECL even received lawsuits from affected patients.[3]

The major components in the software which caused the accidents are the following: stored data, a scheduler, a set of tasks, and interrupt services. An important note in the documentation is that the software allows concurrent access to the shared memory and the set operations for shared variables are not atomic. One of these tasks is the treatment monitor task, which controls all the actions of a live treatment. This task has eight subroutines, and we will only be concerned about three of them: *Datent* (data entry), *Magnet*, and *Ptime*. Datent communicates with another task which handles input from the keyboard using a shared variable to see whether the treatment data has been entered. Once the treatment data has been entered, Datent will signal the machine to start setting up the treatment. The problem of concurrency starts here because the machine uses the cursor on the screen being off the command line to signify the treatment info is being edited, but the shared variable that Datent sets does not ensure the cursor is on the command line before the signal to start the treatment setup. When Datent determines the treatment data is set, it calls Magnet to set the bending magnets which takes about eight seconds and Magnet calls Ptime to essentially sleep for eight seconds while this is happening. However, if data is changed during this time frame (specifically if some mode or energy level is changed), then this causes Ptime, and subsequently Magnet to exit back to Datent which sets up a race condition between the software and hardware.

The Therac-25 incident highlights the importance of formal documentation, verification and validation, and integration testing. AECL stated there was fairly limited documentation for this project despite even using a custom operating system for this machine specifically, and the code was not verified by a third party. The testing that was done is likely insufficient for critical software like this, as the testing was done separately between the software and hardware, as such, did not encounter the problem of race conditions between the two, as this separation of concerns assumes the other components of the product are flawless. Other possible preventions could’ve been taken from their previous models, as those had safety locks to stop operation if anything in the software malfunctioned and trusted the operator to resolve the issue, while the Therac-25 relied completely on software operation.

# References

1. Clark, Ralph W., and Alice Darnell Lattal. “The ethics of sales: finding an appropriate balance." *Business Horizons*, vol. 36, no. 4, July-Aug. 1993, pp. 66+. *Gale In Context: Biography*, link.gale.com/apps/doc/A13244137/BIC?u=utoronto\_main&sid=bookmark-BIC&xid=0907ac2a. Accessed 20 May 2023.
2. Letendre, Linell A., and Martin L. Cook. “Right Vs. Right: Personal Beliefs Vs. Professional Obligations.” *Parameters (Carlisle, Pa.)*, vol. 48, no. 1, 2018, pp. 7–14, <https://doi.org/10.55540/0031-1723.2846>. Accessed 20 May 2023.
3. Leveson, N. G., and C. S. Turner. “An Investigation of the Therac-25 Accidents.” *Computer (Long Beach, Calif.)*, vol. 26, no. 7, 1993, pp. 18–41, <https://doi.org/10.1109/MC.1993.274940>. Accessed 20 May 2023.
4. “Software Engineering Code of Ethics and Professional Practice: IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices.” Science and Engineering Ethics, vol. 7, no. 2, 2001, pp. 231–38, <https://doi.org/10.1007/s11948-001-0044-4>. Accessed 20 May 2023.