CS523 - FINAL

From a purely ethical/moral point of view, how would you apportion blame to each of the parties mentioned in the story? Explain your answer. Preferably, cite sources mentioned in class. Be sure to address all the relevant parties.

Below is the list of people closely involved in the incident. I believe each of them have contributed to the death of the robot operator. Some very significantly than others. I have described below their actions in brief and my opinions about the same.

Randy Samuels, programmer.

He did a fatal programming error by misinterpreting the robot dynamics formulas which set off an exceptional condition which needed manual intervention, namely the entering of the command codes to stop the robot. The evidence against him includes three similar handwritten formulas, scrawled on a piece of yellow legal pad paper. Each formula describes the motion of the robot arm in one direction: east-west, north-south and up-down. Comparing these formulas with the actual program code written by Randy Samuels, it is clear that he misinterpreted the y-dots in the formulas as y-bars and made the same mistake for the x's and the z's: he was supposed to use the derivatives, but he took the averages instead.

He was working under a lot of pressure and because of his arrogant attitude towards his peers he did not take any advice or help from them and hence did not realize that his interpretation of the handwritten formulas was wrong. He should have reached out to his peers and worked with team spirit.

Cindy Yardley, Silicon Techtronics employee and software tester.

A message from Cindy Yardley to Robotics Division Chief Ray Johnson indicates that she faked the test results at his request:

To: ray.johnson
From: cindy.yardley
Re: samuels software
I have finished creating the software test results for that troublesome robot software, as per your idea of using a simulation rather than the actual software. Attached you will find the modified test document, showing the successful simulation.

Should we tell Randy about this?

- *Cindy*

She ran a simulation instead of actual software and broke the ethics of professional competence and integrity by faking the test cases and withholding such critical information from co-workers and customers. It can be inferred from the internal emails that she was forced into faking the test results by her manager Ray Johnson only to save the jobs of the co-workers. Software testing is a very important phase in software development life cycle, its whole purpose is to identify bugs before the product is delivered to the customers. Ideally Cindy should have informed Randy and get the issues fixed. But from the interview with Yoder, we know that she was straight out of school and did not have much experience to develop as a professional and an ethical person. Being an inexperienced professional is why she was easily convinced by Ray that no one would probably ever see the test results.

Sam Reynolds, CX30 Project Manager.

Reynolds was very reluctant to manage a project that would not use the waterfall model, which had served him so well in data processing:

"He attacked agile development as a fad at the meeting on December 11, and after a few verbal exchanges back and forth things got pretty personal. Anderson was especially vocal. She had lots of experience with user interfaces, and from her perspective the operator-robot interface was critical to the success of CX30 since operator intervention would be frequent and at times critical."

Jan Anderson also commented on this aspect of the December 11 meeting:

"Reynolds was vehemently opposed to wasting time -- to use his words -- on any kind of formal analysis of the user interface and its human factors properties. To him, user interfaces were a peripheral issue. Anything new was a fad to him. Computer interfaces were a fad, object-oriented design was a fad, formal specification and verification techniques were a fad, and most of all, agile was a fad."

Robbie CX30 project was supposed to interact with its operator unlike its previous models and hence should have been implemented using prototyping model wherein interface can be designed at an early stage before agreeing on the system requirements. If this was done instead of waterfall model, issues with the robot would have been identified at an early stage of development. Instead, Sam fired Jan Anderson who seemed to have prior experience of working on similar projects. He should have listened to the experts in the domain since he was from data processing background and had limited knowledge on robotics. He broke the ethics of Duty to profession[1].

Also, there was an enormous amount of friction between Sam Reynolds and robotics division chief Ray Johnson:

"They hated each other's guts. By June of last year the robot project had fallen six months behind schedule, and Johnson went through the roof. There were rumors that the entire robotics division, which he headed, would be terminated if the Robbie CX30 project didn't prove a commercial success. He called Sam Reynolds into his office, and he really chewed Sam out. You could hear the yelling all the way down the hall. Johnson told Sam to finish Robbie by the first of January or heads would roll."

The above passage from the news article shows that Sam was under tremendous pressure and was threatened to finish the project on time. Thus, perverse incentives and corporate warfare made Sam Reynolds take wrong decisions right from the beginning.[2]

Another mistake that Sam did was not allocating enough bandwidth to train the new employees. The new programmers were not fully integrated into the project even six months later as a result. He was cutting the corners as mentioned in the below lines from the article:

"I'm not saying that Johnson was ordering Sam to cut corners. I think the idea of cutting corners was implicit. The message was, cut corners if you want to keep your job."

Ray Johnson, Robotics Division Chief at Silicon Techtronics.

Earlier versions of Robbie, the CX10 and the CX20, were experimental in nature, and no one had expected them to be commercial successes. In fact, the robotics division of Silicon Techtronics has operated heavily in the red since its inception six years ago. If the CX30 failed, Silicon Techtronics was going to drop out of the industrial robotics business altogether. Ray Johnson was driven by the division's need for success. Insiders consistently portray Johnson as a manager in desperate need of a successful project. Twenty new programmers were added to the Robbie CX30 project. This was just several days after the stormy meeting between Johnson and Reynolds. Bringing emotional and personal feud into a professional environment is not good. The new hires were a disaster:

"Johnson unilaterally arranged for these new hires, presumably by shifting resources from other aspects of the Robbie project. Reynolds was vehemently opposed to this. Johnson only knew about manufacturing hardware.

[1] Spafford, Eugene H. "Ethics PPT". Slide no.15

[2] Spafford, Eugene H. "Economic Factors PPT". Slide no.10

That was his background. He couldn't understand the difficulties that we were having with the robotics software. You can't speed up a software project by adding more people. It's not like an assembly line."

He worked extensively on manufacturing hardware. His ideologies like: *“Perfect software is an oxymoron,” “Perfection is the enemy of the good”* shows his incompetence as a software manager and his failure to understand the requirements and his team.

The biggest mistake he did was to use his authority to manipulate Cindy which according to me lead to the unfortunate incident. He failed at creating and following protocols in the form of checks and balances. He broke all kinds of codes: professional, ethical, social, legal and as per me is responsible for allowing the Robbie CX30 to go out the door without proper due diligence.

Michael Waterson, President and CEO of Silicon Techtronics.

Sam Reynolds was moved from the data processing division, which takes care of inventory and payroll, to the robotics division just three weeks before the December 11 meeting. He was transferred by Silicon Techtronics president Michael Waterson as per following lines from the news article.

"Waterson thought it would be cheaper to move Reynolds to robotics rather than try to find a new manager for the Robbie project from outside. Also, Waterson tended to be suspicious of people from the outside. He often sends down memos about how long it takes people to master the Silicon Techtronics way of doing things. In Waterson's view, Reynolds was a manager and he was moved to his new position in robotics as a manager and not as a technical expert. Clearly, Reynolds saw himself as both a manager and as a technical expert. Reynolds was not aware of his own technical limitations."

By placing Reynolds in charge of the CX30 project as a cost-saving measure, rather than hiring an expert from outside, Waterson was going against the advice of the robotics division chief Ray Johnson, who strongly opposed the choice of Reynolds to head the Robbie CX30 project.

Discuss the economic issues that led to the outcome. Specifically, refer to economic problems and pressures we covered in the class or the readings to discuss how the parties (and organizations) involved operated. You must include appropriate citations to external references and class slides.

Economic factors like misaligned/perverse incentives, network externalities and information asymmetry played major role in how resources like people time and money were used in CX30 project.

A significant difficulty in the optimal development of technology is the imperative to integrate economic implications into technical designs. But, if a technology requires the party with the least risk make the greatest investment, then that system will fail. The same happened in case of CX30 project which led to a fatal outcome. Robotics division of Silicon Techtronics has operated heavily in the red since its inception six years ago. If the CX30 failed, Silicon Techtronics was going to drop out of the industrial robotics business altogether. Even though Ray Johnson was from manufacturing background, he was made in charge of robotics project in desperate need of a successful project. He was not the right person to manage a software development project and thus began the downfall of the Robbie CX30 project. Ray Johnson was driven by the division's need for success which is nothing but perverse incentive. Since he was not the one operating the robot, he did not care about the safety measure required before delivering the robot. He made the biggest mistake of going with waterfall model and not the prototyping model. All he cared about was finishing the project on time and save his job. Economists call this a moral-hazard effect[3]. Moral hazard is a situation in which one party engages in risky behavior or fails to act in good faith because it knows the other party bears the consequences of their behavior. Legal theorists have long known that liability should be assigned to the party that can best manage the risk. In this scenario, Ray Johnson was not the right fit for the role of managing such high-risk project. Thus, perverse incentives[4] played a major role in the death of the robot operator.

Another factor to analyse here is the competitive corporate warfare and the need for Silicon Techtronics to deliver a revolutionary industry robot to secure a dominant place in industrial robotics business. The earlier Robbie robots got a lot of press. Robbie CX30 was going to capitalize on the good publicity generated by the earlier projects. CX30 represented a gigantic step forward in terms of sophistication. If the CX30 did not succeed, Silicon Techtronics was going to drop out of the industrial robotics business altogether. This is where network economics comes into picture. Software industries are characterized by many different types of externalities, where individual's actions have side-effects on others. In this case, it had the worst side-effect possible, killed an innocent man. Externalities should be considered while allocating the budget for a particular project to ensure proper due diligence before the product delivery. Externalities are a cost or benefit of an economic activity experienced by a third party which has no relation with the one causing the activity. While a product vendor is building market dominance, they commonly ignore safety/quality in the beginning as they are building their market position. Later, once they have captured a lucrative market, they add excessive features in order to lock their customers in tightly.

If you're a manufacturing plant, one way to maximize profit is to keep costs as low as possible. One way to do that is to cut corners. Go ahead and dump that toxic waste into the river and pollute the heck out of the air with your smokestacks. These options are much cheaper than installing smoke scrubbers or trucking waste to proper disposal sites. Of course, economists have long known that this does not paint the entire picture. Taking these shortcuts incur other costs, it's just that these costs are not borne by the manufacturing plant. The term externalities describe such spill over costs. In economics an externality or spill over of an economic transaction is an impact on a party that is not directly involved in the transaction. In such a case, prices do not reflect the full costs or benefits in production or consumption of a product or service. Thus, the full cost of manufacturing

[3] Anderson, R. J. 2001 Why information security is hard—an economic perspective. In Proceedings of Seventeenth Annual Computer Security Applications Conf., December,2001, pages 358–365.

[4] Spafford, Eugene H. "Economic Factors PPT". Slide no.10

includes the hospital bills of those who get sick by drinking the tainted water, the cost of the crops damaged by the acid rain, etc.

Ray Johnson believed “Perfect software is an oxymoron” and was ok with delivering a low quality, even a faulty product. As Robert Glass pointed out in Facts and Fallacies of Software Engineering book, research shows that maintenance typically consumes from 40 to 80 percent of software costs, typically making it the dominant life cycle phase of a software project. Thus, solutions need to factor that in, or the customers will continually be left with the clean-up duty while the polluters have long since moved on.

Most commercial software contains design and implementation flaws that could easily have been prevented. Although vendors are capable of creating more secure software, the economics of the software industry provide them with little incentive to do so. In many markets, the attitude of ‘ship it Tuesday and get it right by version 3’ is perfectly rational behavior. Consumers generally reward vendors for adding features, for being first to market, or for being dominant in an existing market – and especially so in platform markets with network externalities[5]. These motivations clash with the task of writing more secure software, which requires time-consuming testing and a focus on simplicity.

Another aspect of vendors’ lack of motivation is explained by Anderson: the software market is a ‘market for lemons.’[6] by George Akerlof. The software market suffers from the same information asymmetry. Vendors may make claims about the quality of their products, but buyers have no reason to trust them. In many cases, even the vendor does not know how good its product is. So, buyers have no reason to pay more for more safer products, and vendors are disinclined to invest in quality assurance. System reliability can depend on the sum of individual efforts, the minimum effort anyone makes, or the maximum effort any makes. Program correctness can depend on the weakest link (the most careless programmer introducing a vulnerability) while software validation and vulnerability testing might depend on the sum of everyone’s efforts. A simple model by Varian[7] provides interesting results when players choose their effort levels independently. For the total-effort case, system reliability depends on the agent with the highest benefit-cost ratio. In the weakest-link case, the agent with the lowest benefit-cost ratio dominates. As more agents are added, systems become increasingly reliable in the total-effort case but increasingly unreliable in the weakest-link case[8]. What are the implications? One is that software companies should hire more software testers and fewer (but more competent) programmers. This is something that went considerably wrong in case of Silicon Techtronics when Johnson unilaterally added 20 new programmers to the CX30 project presumably by shifting resources from other aspects of the Robbie project, other aspects being quality assurance.

[5] Spafford, Eugene H. “Economic Factors PPT”. Slide no.5

[6] George A. Akerlof. The market for lemons: Quality uncertainty and the market mechanism. Quarterly Journal of Economics, pages 488–500, August 1970.

[7] H. Varian, Economics of Information Security, L. J. Camp, S. Lewis, eds. (Kluwer Academic Publishers, 2004), vol. 12 of Advances in Information Security, pp. 1–15.

[8] Spafford, Eugene H. “Economic Factors PPT”. Slide no.25

Suggest how the economic issues you discussed in #2 could have been changed, resulting in a better outcome for (at least) Bart. Explain why you believe those changes should work.

We discussed earlier how wrong incentives make certain people in authority to take bad decisions like how Sam Reynolds did by choosing a bad software development model which resulted in an unfortunate fatal outcome. One way this could have been avoided is by incentivizing people in charge to make right decisions[9]. The second is to institute policies that discourage an immoral behavior by making it a punishable offense. Incentives maybe in the form of rewards program or by acknowledging the efforts behind any achievements and bringing forward certain accomplishments to everyone's notice. Incentivizing also helps reduce moral hazard.

Cost cutting is another issue seen most where appropriate budget is not allocated to the projects, this could be resolved by including negative externalities in the marginal cost of product development. Negative externalities can be prevented by change in ideologies and combined efforts of all parties involved, not just the one hit by the negative outcome.[10]

We have seen how information symmetry is another reason which can cause moral hazard. This can be resolved by signalling. Vendor can give signals in form of advertising and inform the customers about the quality of the product and why he should buy it. Trial versions of software or applications can be provided based on needs of the customer. This can also help in the design phase where inputs from the customer can also be incorporated into the final product. We have seen beta versions of applications and software help get a validation and feedback directly from customers. Extreme case of signalling can be seen in the form of open-source software.[11]

We saw in the previous discussion that system reliability can depend on the sum of individual efforts, the minimum effort anyone makes, or the maximum effort any makes. Program correctness can depend on the weakest link (the most careless programmer introducing a vulnerability) while software validation and vulnerability testing might depend on the sum of everyone's efforts. We also established a solution for this: that software companies should hire more software testers and fewer (but more competent) programmers.[12]

Above measures if taken would ensure right decisions were made during every step of software development life cycle and Robbie would be tested thoroughly before Bart would operate the robot.

[9] Spafford, Eugene H. "Economic Factors PPT". Slide no.10

[10] Spafford, Eugene H. "Economic Factors PPT". Slide no.12

[11] Spafford, Eugene H. "Economic Factors PPT". Slide no.16-20

[12] Anderson, R. J. 2001 Why information security is hard—an economic perspective. In Proceedings of Seventeenth Annual Computer Security Applications Conf, December,2001, pages 358–365.

Explain if your suggestions in #3 would likely be adopted in the current market and why or why not. Cite articles about specific examples in the market to justify your answer.

The adoption of above-mentioned solutions to economic issues faced in software development are dependent on following factors:

Market competition: The software industry is highly affected by network externalities, and companies are constantly under pressure to release new products and updates to stay ahead of the competition. The dominant firms like Microsoft and Windows create a Lock-in where-in new products and software have a tough time overcoming the consequences of Lock-in[13], especially the developers, they have to use certain software and products which are dominant. In such situations there is very less freedom for innovation and good quality.

This can sometimes result in a focus on short-term goals rather than long-term solutions, which may make it challenging for companies to invest in software testing and validation, or to allocate appropriate budgets for software development. However, if a company recognizes the value of investing in these areas, it could differentiate itself from the competition and potentially gain a competitive advantage.

Economic incentives: The economic incentives for a company to adopt these solutions can vary based on the company's goals and objectives. For example, a company that values its reputation and customer satisfaction may be more willing to reward their employees with bonuses and salary bump. On the other hand, a company that prioritizes cost-cutting may be less willing to do so.

Company culture: The culture of a company can also play a significant role in the adoption of these solutions. For instance, a company that values employee well-being and ethical practices may be more likely to incentivize individuals to make the right decisions, whereas a company that prioritizes profit over ethics may not be as inclined to do so[14].

[13] Spafford, Eugene H. “Economic Factors PPT”. Slide no.10

[14] Spafford, Eugene H. “Economic Factors PPT”. Slide no.23

Were there psychological biases involved that influenced the outcomes? Discuss, and be sure to cite slides and references as appropriate.

Neglect of probabilities[15]

People completely ignore probabilities, instead they focus entirely on the outcomes' magnitudes. This is what most of the team members of Robbie CX30 project did. Everyone, right from the chief Ray Johnson to the just-out-of-the-school program tester Cindy neglected the quality of the product and focused on the fact that if they do not deliver this project on time, then the whole division would be shutdown and they all would eventually lose their jobs. Everybody worked under this huge pressure and ended up making wrong decisions which led to the death of the robot operator.

Availability Heuristic[16]

Jan Anderson was recently fired and that kept every team member of CX30 project convinced that if they do not follow the timeline they would be fired too. Hence, even Cindy gave in to the blackmail that Ray did and faked the test results.

Hyperbolic discounting[17]

All the relevant parties involved in the CX30 project were so focused on getting the project done by the given deadline that they neglected the quality of the product. They allowed a faulty piece of code be delivered to the customer which ended up killing someone.

Status quo bias[18] and Overconfidence[19]

Sam Reynolds clearly displayed this bias when he was so adamant at using waterfall model for CX30 project, going against the domain expert Jan Anderson, who suggested prototyping model. He had always worked with waterfall model in data processing projects and believed that it always works best. He was not open to new ideas and models even when it was very much needed.

Valence effect[20]

This is the tendency to over-estimate the likelihood of favorable outcomes. Cindy said the following about faking the test result:

"*He assured me that no one would probably ever see the test results because the robot was perfectly safe. It was just an internal matter, a matter of cleanliness, in case anyone at Cybernetics or higher up in the corporation got curious about our test results. I asked him whether he was sure about the robot being safe and all that and he said, 'It's safe! The user interface is our line of defense. In about six months we can issue a second version of the robotics software and by then this Samuels problem will be solved.*"

Ray Johnson clearly overestimated the robot to be safe.

Wishful thinking[21]

Randy believed that he was a good at coding and always assumed his code is correct. He can never go wrong and hence did not take help from peers to review his code. If he had discussed the robot's formulas, he might have identified the issue as part of code review process.

[15] Spafford, Eugene H. "Psychological biases, heuristics, and errors PPT". Slide no.10
[16] Spafford, Eugene H. "Psychological biases, heuristics, and errors PPT". Slide no.19
[17] Spafford, Eugene H. "Psychological biases, heuristics, and errors PPT". Slide no.22
[18] Spafford, Eugene H. "Psychological biases, heuristics, and errors PPT". Slide no.24
[19] Spafford, Eugene H. "Psychological biases, heuristics, and errors PPT". Slide no.34
[20] Spafford, Eugene H. "Psychological biases, heuristics, and errors PPT". Slide no.25
[21] Spafford, Eugene H. "Psychological biases, heuristics, and errors PPT". Slide no.35

Correlation related fallacies and Survivorship bias[22]

Ray Johnson's Ivory theory is an example of spurious correlation. Ivory Snow being 99 and 44/100 percent pure was not something that can be compared with robotics software.

Above corelation lead him to believe "Perfect software is an oxymoron" thereby ended up taking wrong decisions.

Social bias[23]

Abandoning one's own belief or information under the influence of other people's actions is social bias. Cindy abandoned her own beliefs and faked the test results just because Ray Johnson convinced her that the robot was safe.

Omission bias [24]

Ray forced Cindy to fake the test results and she believed she had no other choice but to follow since she wanted to keep her job. But Sam was not directly threatened, still he chose to remain silent even after knowing the test results were fake.

[22] Spafford, Eugene H. "Psychological biases, heuristics, and errors PPT". Slide no.39 and 46

[23] Spafford, Eugene H. "Psychological biases, heuristics, and errors PPT". Slide no.53

[24] Spafford, Eugene H. "Psychological biases, heuristics, and errors PPT". Slide no.82

Legal issues are raised – both civil and criminal – in the story. Do they seem reasonable? Explain why or why not. Discuss what changes you might suggest in law that would help address the situation in the book. Again, cite class slides and external references to help support your remarks.

In the news article "Silicon Techtronics Promised to Deliver a Safe Robot" we see that a group of programmers who call themselves the Justice for Randy Samuels Committee raised an issue that it was Silicon Techtronics that was legally bound to deliver a safe robot to Cybernetics, and it was wrong to charge Randy with manslaughter. They also made public the requirements document to prove that Randy was not legally responsible for the death of Bart Matthews.

According to me this seems reasonable as Randy never signed this contract himself with the customer. Moreover, the Robbie CX30 robot was designed to be an "intelligent" robot capable of operating in a variety of industrial settings. Separate requirements documents were necessary for each corporate customer because the Robbie CX30 was not an "off-the-shelf" robot but a robot that needed to be programmed differently for each application. However, all requirements documents for the Robbie CX30 project, including the agreement between Silicon Techtronics and Cybernetics, contain the following important statements:

"*The robot will be safe to operate and even under exceptional conditions (see Section 5.2) the robot will cause no bodily injury to the human operator... . In the event of the exceptional conditions which potentially contain the risk of bodily injury (see Section 5.2.4 and all of its subsections), the human operator will be able to enter a sequence of command codes, as described in the relevant sections of the functional specification (see Section 3.5.2), which will arrest robot motion long before bodily injury can actually occur*."

Exceptional conditions include unusual events such as bizarre data from the robot sensors, erratic or violent robot motion or operator error. It was exactly such an exceptional condition that led to the death of Bart Matthews. These passages were extracted from the portion of the requirements document that deals with "non-functional requirements." These present in complete detail the constraints under which the robot will operate. For example, the requirement that the robot be incapable of harming its human operator is one such constraint, and Silicon Techtronics, was legally obligated to satisfy this constraint.

Thus, even though Randy's error caused the incident, the death of the operator could have been avoided if the user interface was intuitive enough or if the operators were trained what to do in such situations.

Second legal issue raised in the story is regarding the copyright issue. One of the internal e-mails shows that Randy Samuels stole some of the software that he used in the killer robot project. This was revealed in a message Samuels sent to Yardley when she first tested his software and got erroneous results:

In-reply-to: cindy.yardley

From: randy.samuels

Re: damned if I know

I cannot for the life of me figure out what is wrong with this function, swing_arm(). I've checked the robot dynamics formula over and over again, and it seems to be implemented correctly. As you know, swing_arm() calls 14 different functions. I lifted five of those from the PACKSTAT 1-2-3 statistical package verbatim. Please don't tell a soul! Those couldn't be the problem, could they?

- Randy

Copying code from a commercial software package such as PACKSTAT 1-2-3 is a violation of copyright and clearly is a valid legal issue.[25]

[25] United States Copyright Office. Software-Enabled Consumer Products. Dec 2016.

Third legal issue raised in the story is related to the internal email transcripts that were made public. Eavesdropped on private electronic mail communications is very unethical and also a criminal offense[26].

Any law should be designed with prevention as the first priority, in an effort to stop tragedy before it happens. In this situation, Randy Samuels was not likely considering that the software he was designing would ever have the opportunity to injure, let alone fatally wound. That is not the case with a firearm, which is designed mainly to inflict harm, whether for malicious intent or for protection. The last article in the sequence of readings describes it well. The responsibility lies with Samuels, and several others, but not guilt in the legal sense of the word. As such the law should demand strict testing procedures to reduce the likelihood of such tragic events.

[26] Email as Public Record: Five Things You Should Know by Frayda Bluestein Feb 2017, Coates' Canons NC Local Government Law.

The book was initially written in 1997, with the Internet becoming commercial in 1996. Provide some discussion of whether you believe Epstein’s story could actually occur, in some form, today. Cite some supporting evidence from news or magazine sources, journals, etc.

“Robotic surgery linked to 144 deaths in the US” in BBC, 22 July 2015.

“Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian” The Guardian, Sam Levin and Julia Carrie Wong in San Francisco, 19 Mar 2018.

“11 more crash deaths are linked to automated-tech vehicles” by Tom Krisher, LA Times, Oct 18, 2022.

“Death by robot: the new mechanised danger in our changing world” Ian Tucker, The Guardian, 25 Mar,2018.

“Use of police robot to kill Dallas shooting suspect believed to be first in US history” Sam Thielman, The Guardian, 8 Jul 2016.

Above news articles show that the number of deaths by robots and autonomous vehicles is on the rise. As the use of autonomous machines has increased in society, so too has the chance of robot-related fatalities. The real issue is not about the number of deaths robots have caused; rather, it’s about helping the general population have a better understanding of robots. For most people, their understanding of robots comes from the movies. That’s also where their fear of robots developed. For example, from the movie The Day the Earth Stood Still to the modern remake of the TV series Battlestar Gallactica, Hollywood tends to portray robots as human killers determined to destroy the earth. In reality, nothing could be further from the truth. Whether we are talking about robots or any other technology, one thing is certain: It’s not the tool; it’s how you use it. In other words, you can use technology to give you cancer or to cure your cancer. Whether a technology tool or a robot is used for good or evil, it all comes down to the human responsible for creating or controlling it. Humans cause robot-related fatalities.

Another perspective of this is, the automation of U.S. manufacturing — robots replacing people on factory floors — is fuelling rising mortality rate among America’s working-age adults, according to a new study by researchers at Yale and the University of Pennsylvania[27]. For decades, manufacturers in the United States have turned to automation to remain competitive in a global marketplace, but this technological innovation has reduced the number of quality jobs available to adults without a college degree.

My take from these readings is, if not direct incident like the one mentioned in the presented book by Epstein, there are multiple other indirect scenarios discussed above which are causing deaths in general with increasing use of technology and autonomous software without due deligience.

[27] Robots are increasing mortality among U.S. adults, By Mike Cummings, February 24, 2022, Yale News.

ChatGPT has recently emerged to join autonomous vehicles in the marketplace. Citing the context presented by the book, discuss how we should place responsibility when those systems misbehave.

In the presented book, the following discussion between the interviewer and Yoder highlights how we should place the responsibilities in such scenarios:

"Yoder: In my mind, the issue of individual versus corporate responsibility is very important. The corporation created an environment in which this kind of accident could occur. Yet, individuals, within that system, acted unethically and irresponsibly, and actually caused the accident. A company can create an environment that brings out the worst in its employees, but individual employees can also contribute to the worsening of the corporate environment. This is a feedback loop, a system in the classical sense. Thus, there is some corporate responsibility and some individual responsibility in the case of the killer robot."

Whether it be an autonomous vehicle or a robot, the responsibility lies with the humans who built them. Same goes with chatGPT. If a doctor is referring to chatGPT for medical treatment, then it is his responsibility to not blindly rely on the AI as there can be symptoms not known to the machine. If something goes wrong with the diagnosis made by chatGPT, the doctor cannot go blame the creators of chatGPT. It just stores what was fed to it and cannot think on its own. Ultimately, it's the responsibility of the doctor or any user in general to use the information provided by the GPT carefully and not misuse it.