Case Study 1: A Breach in Security

In this case study, confidential data collected from a city’s homeless shelters and free clinics was compromised when an employee laptop was stolen. This situation violates a number of ACM ethics principles. First, it is a clear violation of principle 1.2. While the initial collection of data may have been done in an effort to help people, this data being compromised has the potential to lead to identity theft, among other potential outcomes, which may limit people’s access to shelters and clinics. This case study is also a clear violation of principle 1.6. Information that is above and beyond what should be required appears to have been systematically collected, specifically social security numbers. Additionally, this information was not adequately protected in a way that ensures stakeholder confidentiality. The primary ethics failures concern principles 2.5 and 2.9. There were inadequate safeguards in the event non-employees gained access to the laptops that stored the data. And an employee losing a mobile device is a high-likelihood event and is something that should have been accounted for in the design and review of the system.

Thankfully there are a number of measures that could be implemented to reduce the damage that this situation could cause in the future. The first is to minimize the data collection to only what is absolutely necessary. This would reduce the amount of potentially damaging data that could be compromised by future breaches. A further improvement could be made by heavily encrypting confidential data, making it more difficult to exploit in the event of unauthorized access. An additional option is to not store the data locally on employee devices, but rather have the devices connect to the database. This data could then be stored more securely on site or in the cloud. If this were implemented, a lost or stolen laptop could simply be blocked from accessing the database. This could be further enhanced by requiring multi-factor authentication in order to access the database. One final improvement would be to fully anonymize the data that employees have access to. It is unlikely that social security numbers are routinely required by employees, so this more sensitive data could be kept in a separate database, with individuals being differentiated in the main database by a simple ID number.

Case Study 2: Occidental Engineering Case Study: Part 1

In this case study, a company ships air traffic control software with a known, potentially deadly defect, without informing the customer. The engineer who knew about the bug was convinced to keep quiet while a fix was implemented. While nobody died in this hypothetical retelling, several ethics principles were violated. The main violations are of principles 1.1 and 1.2. Hiding this known defect from the customer had a significant chance of being disastrous. Mid-air aircraft collisions will almost always result in the deaths of everyone on board, and a single incident could result in hundreds of deaths. Additionally, the company’s plan was to quickly fix this glitch, but there is no guarantee that they would have been able to do that, or that they would have even been in business long enough to do so. Florida has had prior experience with a culture and management brow-beating engineers into accepting risk, and it resulted in the Challenger disaster. This hypothetical company was very lucky that this didn’t result in disaster. This case study also presents a clear violation of principle 1.3. The software was presented as if it was fully functional, despite knowing that it wasn’t. The fact that this software was safety critical makes this violation even worse than it already is. And while there are several other principles that have been violated, I also want to discuss violations of principles 4.1 and 4.2. The engineer knew the danger, they knew what they were doing was wrong and had the potential to get people killed, but they went along with it anyways. I understand the pressure that they were facing, both from management and from their own team, to just ship the product, but it was clearly the wrong decision.

There are two main ways to approach this problem, and they can be attempted sequentially. The first thing to do is to simply explain that there is still a bug that needs to be worked out and more time is required. If that request is denied, then the customer needs to be told about the defect. In the worst-case scenario, this may result in the contract being cancelled and the team being fired, but this is greatly preferable to the worst-case scenario of shipping the software, which is hundreds of deaths, and then everyone at the company still loses their jobs. If what the company says about the testing process is true, then the software may still be accepted for small scale testing, with the understanding that the defect will be patched before the full contract is paid out. If the customers response is to cancel the project, then the engineers involved can still walk away with their conscious intact, which really is the only thing that will matter when they look back on their life 50 years down the line.