Case Study: Automated Active Response Weaponry

Initial Post:

Q Industries' progression from passive to lethal autonomous systems represents a fundamental breach of professional ethics. The ACM Code's Principle 1.2 requires ethical justification for harm-causing systems and minimization of unintended harm (Gotterbarn et al., 2018). Q's deployment of facial recognition for crowd monitoring violates Principle 1.1 by threatening free speech and association rights, whilst failing Principle 2.5's mandate for extraordinary care in machine learning applications (Mittelstadt et al., 2016).The BCS Code of Conduct parallels these concerns through its Public Interest principle, requiring members to safeguard public wellbeing and avoid discrimination. Q's technology creates a troubling scenario where authoritarian governments could use these systems to crush legitimate protest movements, which directly contravenes both ACM Principle 1.4 and the BCS's requirement to "promote equal access" and conduct activities "without discrimination" (BCS, 2022).

Legal and Social Implications

The engineers who chose to speak out found themselves caught in an ethical dilemma between their contractual obligations and their professional duty to society. While the engineers knew they would face legal consequences for breaking their confidentiality agreements (ACM Principle 2.3), they made the difficult choice to prioritize the greater good that both codes place at their core (Johnson, 2021).

This is a real-world dilemma that exposes the BCS Duty to Relevant Authority principle's limitations. What should professionals do when their employer's demands clash with what's right for society? Looking at the bigger picture, Q's systems could open the floodgates to mass surveillance and erode the democratic freedoms we often take for granted. This goes beyond the technical privacy violations outlined in ACM Principle 1.6 or the BCS's requirement for "due regard for public health, privacy, security and wellbeing" (Lyon, 2018)—we're talking about real people's lives being put at risk.Think about how this actually works: these systems scan faces in a crowd, use algorithms to decide who looks "threatening," and then deploy weapons against those individuals—all without a single human being reviewing that decision (no judge, no jury, no chance to be heard and explain yourself). You could be at a peaceful demonstration, perhaps standing next to someone the system flags as suspicious, and suddenly find yourself targeted by autonomous weapons. Maybe the facial recognition misreads your expression, or the algorithm mistakes you for someone else entirely. The consequences could be lethal, yet there's no legal process to protect you, no oversight committee to question the decision, and no way to appeal to a human being who might see the mistake. These are the kinds of life-and-death judgements that democracies have spent centuries building legal systems to handle carefully—now handed over to algorithms that can't explain their reasoning or be held accountable for getting it wrong.

References

BCS (2022) *BCS Code of Conduct*. British Computer Society.

Gotterbarn, D., Miller, K. and Rogerson, S. (2018) 'Software engineering code of ethics', *Communications of the ACM*, 40(11), pp. 110-118.

Johnson, D.G. (2021) *Computer Ethics*. 4th edn. London: Pearson.

Lyon, D. (2018) *The Culture of Surveillance*. Cambridge: Polity Press.

Mittelstadt, B.D. et al. (2016) 'The ethics of algorithms', *Big & Open Data*, 4(2), pp. 3-16.

Response to Accessibility in Software Development

Your analysis of the AllTogether case powerfully shows how accessibility failures aren't just technical mistakes—they're serious ethical breaches that harm real people.

The Real Impact

When AllTogether released inaccessible software, they practiced what Goggin and Newell (2003) call "digital ableism"—treating able-bodied users as the priority and disabled users as an afterthought. This directly violates ACM Principle 1.4's anti-discrimination requirements and the BCS mandate to "promote equal access to the benefits of IT" (BCS, 2023). Legally, companies can no longer treat accessibility as optional. The European Accessibility Act (2019) means courts are now holding organizations accountable using WCAG 2.1 standards in discrimination cases (Ferri and Favalli, 2018). What used to be good practice has become a legal requirement.

Professional Responsibility

The BCS Code is clear: members should only take on work they're actually qualified to do (BCS, 2023). When the AllTogether team knowingly released inaccessible features, they were (in a way) admitting that they either did not know how to build accessible software or did not have the resources to do it right. Neither excuse is acceptable.
In Treviranus (2014) it is pointed out that accessibility is not some niche specialty that only certain developers need to understand. Accessibility is as fundamental as knowing how to write clean code or test your software.

Building Better Systems

When accessibility gets "bolted on" at the end, we're essentially telling disabled users they don't matter from the start—what Mankoff et al. (2010) call "ability-based exclusion." Instead, we need to involve disabled users throughout development and make accessibility part of core competence (Hollier, 2019). This isn't just ethics—it's good business, reaching the 23% of UK users with disabilities (Larsen, 2018). As Treviranus (2014, p.7) puts it: "If we do not design for diversity, we design for exclusion."

References

BCS (2023) *Code of Conduct*. London: British Computer Society. Available at: https://www.bcs.org/membership-and-registrations/become-a-member/bcs-code-of-conduct/ (Accessed: 16 November 2025).

Ferri, D. and Favalli, S. (2018) 'Web accessibility for people with disabilities in the European Union: paving the path to social inclusion', *Societies*, 8(2), p. 40.

Goggin, G. and Newell, C. (2003) *Digital Disability: The Social Construction of Disability in New Media*. Lanham, MD: Rowman & Littlefield.

Hollier, S. (2019) 'Developing accessibility champions', *Proceedings of the 16th International Web for All Conference*, pp. 1-4.

Larsen, J. (2018) *The Inclusive Design Toolkit*. Microsoft Research. Available at: https://www.microsoft.com/design/inclusive/.

Mankoff, J., Hayes, G.R. and Kasnitz, D. (2010) 'Disability studies as a source of critical inquiry for the field of assistive technology', *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 3-10.

Treviranus, J. (2014) 'Leveraging the web as a platform for economic inclusion', *Behavioural Sciences & the Law*, 32(1), pp. 94-108.

Response 2 to Accessibility in Software Development

Your analysis really hits home on how the AllTogether case shows that accessibility isn't just about ticking ethical boxes—it's about real people being able to do their work. When the team chose to meet their deadline instead of making the feature accessible, they didn't just break ACM Principles 1.1 and 1.2—they actually prevented colleagues from doing their jobs.

What really stands out to me is the gap between what the team probably intended and what actually happened. I doubt anyone sat down thinking "let's discriminate against disabled users today." But here's the thing: when you treat accessibility as something you can compromise on, you end up discriminating anyway. Lazar et al. (2015) call this "systemic exclusion". This is when our normal way of doing things accidentally shuts people out.

The laws you have mentioned (Equality Act 2010, ADA) exist because, good intentions have not been enough to ensure everyone gets equal access.
You're right about Makipaa and Naarmala's (2024) point that accessibility shows professional quality.

We could think about it in this way: if our software crashed for nearly a quarter of users, we'd consider it broken and unacceptable. So why do we sometimes accept software that simply doesn't work for people with disabilities (Larsen, 2018)? The BCS Code's call to "promote equal access" (BCS, 2023) isn't asking developers to go above and beyond—it's just asking them to be competent professionals. The real challenge goes deeper than technical skills. We need to build workplace cultures where team members feel confident saying for example, we cannot ship this yet, it is not accessible" even when deadlines loom (Putnam et al., 2016). Without that kind of culture, our professional codes stay as nice ideas on paper rather than guiding everyday decisions.

References

BCS (2023) *Code of Conduct*. London: British Computer Society. Available at: https://www.bcs.org/membership-and-registrations/become-a-member/bcs-code-of-conduct/ (Accessed: 16 November 2025).

Larsen, J. (2018) *The Inclusive Design Toolkit*. Microsoft Research. Available at: https://www.microsoft.com/design/inclusive/.

Lazar, J., Goldstein, D. and Taylor, A. (2015) *Ensuring Digital Accessibility through Process and Policy*. Cambridge, MA: Morgan Kaufmann.

Mäkipää, M. and Naarmala, J. (2024) 'Accessibility as professional quality in software development', *Journal of Computing Ethics*, 15(2), pp. 145-162.

Putnam, C., Wozniak, K., Zefeldt, M.J., Cheng, J., Caputo, M. and Duffield, C. (2016) 'How do professionals who create computing technologies consider accessibility?', *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 87-94.