**Why Software Developers Should Take Ethics into Consideration**

Sankara Narayana Katabathina

Colorado State University Global

Course Code: CSC502

Instructor: Dr Shaher Daoud

Due Date: 08/02/2025

**Why Software Developers Should Take Ethics into Consideration**

Software is at the core of modern life. From transportation systems to healthcare, dating platforms to municipal tax programs, software developers hold immense responsibility for how digital systems impact users and society. With this reach comes a profound ethical duty. Unfortunately, many systems are designed with functionality and profit in mind but without due attention to fairness, privacy, accountability, or harm prevention. Ethical lapses in software development can lead to biased algorithms, compromised data privacy, exclusionary design, and even loss of life.

**The Importance of Ethics for Software Developers**

Ethics in software development is not a luxury it is a necessity. Software developers make design decisions that affect how people interact with digital systems and each other. These decisions often have long-lasting consequences. Developers determine how data is collected, what features are prioritized, how recommendations are made, and how security is enforced.

Consider the case of Clearview AI, a company that scraped billions of images from the internet to build a facial recognition tool used by law enforcement. While technically impressive, the software raised widespread ethical concerns about surveillance, consent, and racial profiling. Technology outpaced legal and ethical standards, putting innocent individuals at risk of misidentification. Developers and executives had followed stricter ethical guidelines such as obtaining consent, bias testing, and transparency, many issues could have been prevented.

Ethics guides developers toward values like user safety, fairness, honesty, and respect. It helps developers anticipate risks, navigate ambiguous situations, and prioritize human dignity over unchecked innovation.

**Examples of Ethical Considerations Developers Might Face**

Software developers routinely encounter ethical dilemmas. Below are examples across various industries:

**1. Privacy and Data Use**

Apps collect vast amounts of user data locations, search histories, contacts, and more. Developers must decide how much data to collect, how long to retain it, and how to ensure consent. For example, genetic testing services like 23andMe have been criticized for sharing anonymized genetic data with third party pharmaceutical firms. Although legally permitted, the ethical issue is whether users truly understand the scope of data sharing.

**2. Algorithmic Bias**

Automated hiring platforms like those used by Amazon have shown bias against women because the training data was based on male dominated historical hiring patterns. Developers may unintentionally encode discrimination if they fail to review and correct skewed data sets.

**3. Dark Patterns**

Some apps manipulate users into sharing more than they want, signing up for subscriptions, or giving up privacy often through deceptive design known as dark patterns. Developers may face pressure to increase engagement at the expense of user wellbeing.

**4. Security Trade Offs**

Developers may be tempted to cut corners on security to save time. The Equifax breach in 2017, which exposed personal data of over 147 million Americans, was partially due to failure to patch a known software vulnerability.

Each of these examples reflects ethical decisions made or ignored during the software lifecycle.

**How Developers Can Incorporate Ethical Considerations**

Ethical thinking must be integrated into each stage of software development:

**1. Requirements Gathering**

From the beginning, developers should ask: Who are the stakeholders? What are the risks of misuse or harm? What values should the system prioritize. example, transparency, accessibility, or fairness?

**2. Design and Prototyping**

Use design principles that avoid bias and respect diversity. Consider building ethical checklists into Agile or DevOps workflows. These checklists could include questions such as: Does this feature benefit all user groups equally? Is user consent explicitly obtained?

**3. Testing and Deployment**

Implement fairness testing, penetration testing, and ethical reviews during QA. Simulate edge cases what happens when the system is used in unintended ways?

**4. Post Deployment Monitoring**

Ethical software development doesn't end with deployment. Monitor the software for misuse, unintended consequences, or evolving threats. Build in mechanisms to issue corrections or decommission problematic features.

**5. Inclusive Design and Accessibility**

Ethical software must work for all users, including those with disabilities or from marginalized communities. Developers should follow Web Content Accessibility Guidelines (WCAG) and design with inclusiveness from the start, not as an afterthought. For instance, colorblind users may struggle with red/green UI elements, and visually impaired users may rely on screen readers. Ethical developers test their software with diverse users and ensure that design choices do not exclude or frustrate anyone based on physical, cognitive, or cultural differences.

**6. Interdisciplinary Collaboration**

Ethics cannot be fully addressed in a technical silo. Developers should collaborate with ethicists, legal experts, social scientists, and stakeholders to better understand the implications of their systems. For example, when designing AI models for public services like education or policing, working with legal scholars and community representatives can surface risks that developers alone may not recognize.

**7. Ethical Use of Open Source and Third-Party Tools**

Many developers integrate open-source libraries or third-party APIs without fully reviewing their origins or ethical implications. It is vital to assess whether these components have known vulnerabilities, controversial uses, or violate user privacy. Ethical development includes choosing tools that align with human rights values and maintaining awareness of the broader supply chain.

**8. Internal Advocacy and Leadership**

Individual developers can lead ethical initiatives within their organizations. This may include forming ethics committees, proposing impact assessments, or documenting design decisions with ethical justifications. Developers who speak up about harmful practices or advocate for fairer features, especially under pressure, play a crucial role in changing company culture.

By embedding ethics across design, testing, accessibility, and collaboration, developers can help ensure that software systems serve users equitably and minimize unintended harm

**What Questions Should Developers Be Asking?**

Ethical software developers should move beyond technical feasibility to ask broader questions, including.

Who might be harmed by this feature or algorithm?

What biases might exist in the training data or system design?

Are we being transparent with users about how their data is used?

What would an average user reasonably expect this system to do?

Are we treating all user groups fairly—regardless of race, gender, ability, or location?

Is there a way to design this feature to be more inclusive or respectful?

Asking these questions can uncover ethical blind spots early and prevent crises later.

**How the Software Community Can Support Ethical Coping**

Individual ethics is not enough. The software engineering community must foster a culture that supports ethical awareness and action. Here are some strategies.

**1. Professional Ethics Codes**

Organizations like ACM and IEEE provide codes of ethics that developers can use as a foundation. These codes emphasize honesty, safety, respect for privacy, and responsibility to society.

**2. Ethics Education**

Computer science programs should teach ethics alongside algorithms and data structures. Courses should include case studies, real world dilemmas, and philosophical frameworks like utilitarianism and deontology.

**3. Open Dialogue and Whistleblower Support**

Companies must create safe spaces for engineers to raise concerns without fear of retaliation. Promoting open discussions around ethical risks before a product launch is critical.

**4. Ethics Review Boards**

Just as Institutional Review Boards (IRBs) monitor human subject research, software companies should establish Technology Ethics Boards to review major system launches for ethical concerns.

With these supports, ethical software development becomes a collective responsibility, not an individual burden.

**Consequences of Unethical Software and Why They Matter**

There will absolutely be consequences for unethical software, because the stakes are often high. Unethical software can ruin reputations, violate rights, and even cause physical harm. Let's consider two examples.

**Uber's Self-Driving Vehicle Crash (2018)**: A pedestrian was killed due to poor safety logic in the AV's design. Uber disabled the system's emergency braking to prevent erratic stops during testing. Developers made decisions that prioritized data collection over safety.

**Cambridge Analytica Scandal (2016)**: The misuse of Facebook user data to influence political opinion during elections shook public trust in digital platforms. Developers failed to enforce data protection protocols and audit misuse.

When there are no consequences, companies may continue to release flawed systems, normalize harm, and silence internal whistleblowers. Consequences, whether legal, reputational, or financial pressure create pressure to improve.

accountability should be balanced. Not every engineer has decision making power. It is important to direct consequences toward those with the authority to enforce ethical standards project managers, executives, or policy leaders while educating and supporting the engineers on the ground.

**Conclusion**

Software developers are more than coders they are architects of modern society. Their work shapes how we communicate, travel, work, vote, and receive healthcare. With this influence comes a deep ethical responsibility. Ignoring that responsibility invites bias, exploitation, exclusion, and harm.

By embedding ethical questions into design processes, building supportive community norms, and advocating for accountability, developers can create technology that respects human dignity and strengthens society. Ethics should not be seen as a burden but as a core competency of professional, high quality software development.

**References**

Bradley, James. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy(2017). <https://csuglobal.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,uid&db=edsgao&AN=edsgcl.497487651&site=eds-live>

Durham, Frank D. Parochialism, propaganda and Public Opinion : Reading Lippmann in Zuboff's Age of Surveillance Capitalism. <https://csuglobal.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,uid&db=aph&AN=163742171&site=eds-live>

Joyce, Douglas James. ETHICS IN THE INFORMATION AGE. <https://csuglobal.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,uid&db=bth&AN=9263894&site=eds-live>

Stein, Krysten. Community Led Practices to Build the Worlds We Need. <https://csuglobal.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,uid&db=edsgao&AN=edsgcl.712300548&site=eds-live>

Yu, Lauren J. SHUTTING OUT NOISE AND UNDERSTANDING ARTIFICIAL INTELLIGENCE. <https://csuglobal.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,uid&db=f5h&AN=179302814&site=eds-live>

Longworth, Jackson. Race After Technology: Abolitionist Tools for the New Jim Code. https://csuglobal.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,uid&db=aph&AN=150400201&site=eds-live