

Critical Thinking Module 3: Ethical Frameworks in Software Engineering

Alexander Ricciardi

Colorado State University Global

CSC502: Ethical Leadership in Software Development

Professor: Dr. Steven Evans

February 1, 2026

Critical Thinking Module 3: Ethical Frameworks in Software Engineering

Generative Artificial Intelligence (AI) is changing how software engineering is done. AI is now mostly implemented and delivered through software applications; in practice, a user interacts with AI as software. This implies that software engineering and AI engineering are increasingly becoming synonymous. Additionally, AI systems are data-intensive systems that produce, collect, manipulate, and are trained on large amounts of data. This dependency creates substantial risks regarding human rights, risks such as privacy violations, population biases, and security that can violate civil liberties and generate inequalities, causing considerable harm to society (NIST, 2023). These risks must be addressed by implementing an ethical framework designed to approach software as more than just a tool, but as an actor having an impact on society. This essay evaluates the Federal Data Strategy's Data Ethics Framework's principles within software engineering by providing relevant examples of how its seven tenets can be translated or mapped into actionable engineering requirements. This paper also argues that software engineering and AI engineering are increasingly synonymous, and treating ethics as a core software quality attribute rather than an optional constraint is essential to prevent ethical debt and ensure that AI-driven systems remain accountable, transparent, and socially responsible.

Software Ethical Frameworks: Decision Frameworks

An ethical framework is a structured way of reasoning and deciding when choices have moral implications (Carnegie Council, n.d.). In other words, ethical frameworks are decision-making tools that use policies, principles, and procedures to help organizations make ethical choices.

In software engineering, this decision framework is particularly important, as software is rarely a single standalone entity but a set of embedded and entangled components that undergo continuous change throughout their lifecycle, such as shifting requirements, changing user behaviors, new data sources, and advancements in information technologies. Furthermore, most software systems today are data-driven; this trend will intensify with the integration of AI models within software applications. By nature, AI is data-intensive, producing, collecting, manipulating, and training on massive amounts of data. This agglomeration of data-driven systems, which are increasingly embedded within software applications, combined with the sensitive nature of the data processed and the dynamic nature of the software's lifecycle, poses substantial human rights risks that include privacy violations, population bias, and security threats. Moreover, if not properly addressed, this accumulation of risk builds ethical debt that will inevitably generate inequalities and infringements on the civil liberties of individuals, ultimately causing significant social harm.

To prevent the accumulation of ethical debt within software systems, an ethical framework needs to be implemented within every phase of the Software Development Lifecycle (SDLC). A decision framework that interrogates the “why” behind every planning, design, and implementation decision to ensure that the software application empowers individuals and society rather than becoming instruments of exploitation, bias, and harm (Ricciardi, 2026). The framework's ethical core incorporates the principle of data ethics, as data is at the core of modern software architecture. Data ethics can be described as a fairly new branch of ethics that focuses on studying and evaluating moral issues related to data (including its production, processing, manipulation, sharing, and use), algorithms (including AI), and the practices that shape how these technologies are built and deployed (Floridi & Taddeo, 2016).

This implies anchoring every phase of a software's lifecycle to an ethical framework based on the core principles of data ethics. The U.S. General Services Administration offers such a framework in its 2020 "Federal Data Strategy's Data Ethics Framework" by establishing a set of guidelines and defining seven essential data ethics tenets for federal agencies (GSA, 2020).

Analysis of the Federal Data Strategy's Data Ethics Framework

The Federal Data Strategy's Data Ethics Framework was developed to encourage responsible data use across federal agencies, and it is described as an ethical guide for leaders and data practitioners to use throughout the data lifecycle (GSA, 2020). The framework integrates a set of ethical values, seven data ethics tenets, case-based examples of AI bias, dissemination of records and impacts, and a set of questions intended to help data practitioners to examine ethical issues and impacts at each phase of the data lifecycle, rather than simply following a task "checking the box." To evaluate this framework in the context of software engineering, the framework is treated as an ethical decision framework encapsulating every phase of the SDLC. This implies that each of the seven tenets is treated as a software constraint and requirement for design, implementation, and deployment. In practice, it is not about whether software engineers can agree with these ethical tenets, but whether these tenets can be translated into software engineering checks and balances, such as requirements, architecture decisions, testing criteria, documentation artifacts, release gates, monitoring, incident response, and post-deployment review.

Tenet-by-tenet Evaluation with Software Engineering Examples

Tenet 1 - Uphold Applicable Statutes, Regulations, Professional Practices, and Ethical Standards mandates following all laws and regulations, and fulfilling professional responsibilities (GSA, 2020).

In software engineering, this implies the concept of the compliance-by-design approach to software development, which states that codes, statutes, regulations, and policy must be implemented into explicit, testable requirements and need to be enforced. However, this tenet is insufficient to prevent the novel harms created by the emergence of AI generative models. Furthermore, most governments lack regulations that can address these issues, such as proxy discrimination, unsafe automation reliance, and vulnerabilities introduced through AI integrations (NIST, 2023). The Artificial Intelligence Risk Management Framework (AI RMF) proposes an ethical approach to evaluating legal and regulatory compliance throughout the SDLC, rather than relying on a one-time approval.

Tenet 2 - Respect The Public, Individuals, and Communities mandates respect for the public and the communities affected by the usage of the data (GSA, 2020). In software engineering, this implies treating software as not just a technical system but also as a social system where design choices can be detrimental or beneficial across various social groups. This requires identifying affected groups, documenting impacts, and building mechanisms that address potential social and ethical issues. Within AI software systems, this requires identifying false negatives, outliers, and whether the system design increases biases against minorities and vulnerable populations. In the context of software engineering, this tenet is similar to the Association for Computing Machinery (ACM) Code of Ethics principle (Principle 1.1: Contribute to society and to human well-being), which mandates prioritizing human well-being and avoiding harm (ACM, 2018). However, Tenet 2 places the responsibility for implementing these ethical controls solely on software developers, who can, in certain circumstances, interpret these controls as purely symbolic measures rather than enforceable safeguards.

Tenet 3 - Respect Privacy and Confidentiality mandates and describes privacy and confidentiality as protections against intrusion and misuse (GSA, 2020). To minimize potential privacy risks and protect sensitive information, this tenet encourages using data protection and management techniques such as data minimization (use only the data needed), data usage limitation (use the data for one purpose), secure processing (ensure that the data processing is secure), access control (who can access what), and retention/deletion (store and delete after use). For instance, an AI customer service chatbot application that stores user conversations must treat these conversations as sensitive and private data by implementing collection ethical control dictating which conversations can be or not be collected, where, and how long collected discussions can be stored, who and how can the stored discussions be accessed, and establish what parts if any stored discussions can be repurposed. This aligns with Floridi and Taddeo's (2016) argument that an ethical framework must evaluate all data interactions among data, with algorithms (AI models or code), and with data manipulation processes. In the context of software engineering, Tenet 3 mandates performing privacy impact assessments, architectural risk analysis, and implementing a privacy-by-design architecture in the SDLC rather than after deployment.

Tenet 4 - Act With Honesty, Integrity, and Humility addresses transparency, self-evaluation, and self-overreliance. This tenet is critical for AI software applications that interact directly with users and data, as they can generate erroneous outputs that may appear corrected and convincing (GSA, 2020). In software engineering, honesty, integrity, and humility can be translated into the practice of documenting the system limitations and risks, such as known possible failures, uncertainty scales for AI models, and the software limitations (NIST, 2023).

For example, for an AI software system, such as an AI agent or Chatbot, Tenet 4 implies designing a system that knows its limitations and risks associated with generative AI, limitations such as the size of a model context window, and risks such as hallucinations. Furthermore, it requires implementing mechanisms that accept responsibility and transparently rectify erroneous responses.

Tenet 5 - Hold Oneself and Others Accountable dictates accountability as a shared responsibility across internal roles within the organization and the external roles, such as data suppliers (GSA, 2020). In software engineering, accountability (who and how) and traceability (where and when) are operationalized by governance controls such as detailed documentation, logs, clearly defined ownership, and open code artifacts, allowing for tracing a program's behavior back to individual developers and data sources. These controls can be reinforced by executing peer review, automated security scanning, approval documentation for changes, and owners' signatures. However, this tenet does not explicitly mandate enforcement mechanisms ensuring that these controls are applied within every phase of the SDLC as recommended by NIST AI RMF, which requires that governance and traceability controls be treated as an ongoing explicit process rather than an after-the-fact auditing process (NIST, 2023).

Tenet 6 - Promote Transparency defines transparency as a condition for trust and oversight (GSA, 2020). In software engineering, transparency can be implemented through documentation and technical disclosures such as data sources, evaluation reports, decision logs, technical specifications, and user operation manuals. For AI software systems, transparency builds trust by requiring the disclosure of the software system's logical functionality, potential biases, and how users and external systems can influence the software's behavior.

However, transparency is context-dependent, meaning that disclosure can increase privacy risk depending on the context. The Organization for Economic Co-operation and Development's AI Recommendation defines a transparent system as a system that is open to oversight while remaining compliant with safety standards, private rights, laws, and regulations (OECD, 2019). This approach is a potential solution to the transparency-security tradeoff.

For example, while high-level system functionalities are disclosed to the public and open to oversight to ensure trust, sensitive algorithmic details and user data are restricted.

Tenet 7 - Stay Informed of Development in Data Management and Data Science
mandates staying informed recognizes that ethical risk evolves with novel and changing technologies, different data sources, and new adversarial techniques (GSA, 2020). In the context of software engineering, this tenet implies that ethics cannot be treated as a one-time thing just during the design phase of the SDLC, but it must be treated as a continuous process, pre- and deployment and during the maintenance phase.

The tenet warns that advanced technologies can introduce hard-to-detect biases and mandates continuous technical vigilance, such as ongoing testing, monitoring, reassessment, and human oversight.

Mapping the Framework to the SDLC and MLOps Lifecycle

In the context of software engineering, including AI systems, the Federal Data Strategy's Data Ethics Framework is effective when its tenets are translated into repeatable checkpoints and artifacts embedded within each of the SDLC and MLOps lifecycle. This matters because AI software system risks are socio-technical: they generate not only from code and AI models, but also from data, developers' governance choices, and pre- and post-deployment.

In other words, the tenets can be translated into lifecycle ethical controls such as legal and stakeholder constraints in planning (Tenets 1–2), privacy and source controls in data processing (Tenets 3 and 5), documentation and ethics evaluation during the software development process (Tenets 4–6), and monitoring plus accountability the SDLC and MLOps lifecycle (Tenets 5 and 7). Additionally, this interpretation aligns with the NIST AI RMF’s view that governance and risk management must persist throughout design, deployment, and operation (NIST, 2023).

The Framework Strengths and Limitations

The framework's main strength is its lifecycle-aware nature; it is designed to implement actional controls throughout the SDLC and post-deployment rather than checkbox compliance (GSA, 2020). It also aligns with recognized AI ethical themes described in the NIST AI RMF and OECD AI Recommendation, which define transparency and accountability as not just being legal requirements but essential ethical components to avoid the accumulation of ethical debt. This alignment is especially useful for software engineering because it allows tenets to be mapped onto governance artifacts or controls such as requirements, test gates, documentation, monitoring, and incident response. Its major limitation is its non-prescriptive nature, meaning the tenets can be acknowledged in principle while the engineering ethical controls remain optional in practice. Additionally, the framework does not provide explicit ethical guidance for advanced AI/ML system applications; instead encourages relying on best ethical practices. This implies the tenets need to be proactively supplemented with explicit testing, evaluation, monitoring, and governance control mechanisms; if not, ethical debt will still accumulate.

Recommendations

To translate this framework into sustainable and enforceable software practices, the tenets should be implemented within mechanisms dictated in software engineering behaviors, such as definition-of-done criteria (does the code compile, and is ethically compliant), release gates (stop, verify, check for biases, safety, and ethical compliance, then go), and operational runbooks (maintain, monitor, troubleshoot, and continuous ethics monitoring). Additionally, the following controls can be directly mapped to or derived from the tenets:

- Convert Tenets 1–3 into explicit nonfunctional requirements (privacy, auditability, contestability, retention/deletion, and stakeholder impact constraints).
- Require provenance records, decision logs, and clear documentation of known limitations, defects, and bias risks.
- Test, evaluation, verification, and validation (TEVV) for ethical compliance before deployment and test regularly in operation.
- Operationalize Tenet 5 by defining clear ownership for datasets, models, and high-impact changes, including defining approval workflows and incident responsibilities.
- Implement transparency that supports oversight while preserving privacy and security, with clear correction procedures and feedback channels.
- Treat Tenet 7 as a commitment by monitoring technical change and innovation, and by implementing ongoing staff training, so “staying informed” is a state rather than a slogan.

Implementing these controls treats ethics as a software quality attribute similar to security and reliability, which will reduce the likelihood of ethical debt accumulation and prevent public harm, legal consequences, and loss of public trust.

Conclusion

The integration of Generative AI into modern software has fundamentally changed how ethical responsibility is handled by the software engineer. AI models are no longer just a data science tool; they are becoming an integral component of software applications, and by design, AI systems are data-intensive. This has pushed the traditional software engineering ethical frameworks to further encompass data ethical principles in their design. The analysis of the Federal Data Strategy's Data Ethics Framework shows that ethical data frameworks provide high-level ethical principles for data processing. In the context of software engineering, the true values of these principles lie in their capacity to be translated into practical engineering artifacts or controls, such as non-functional requirements, release gates, and continuous monitoring protocols. This can be done by embedding the Data Ethics Framework's seven tenets into the SDLC and MLOps lifecycles. The resulting ethical framework allows developers to address the socio-technical ethical issues associated with AI software systems. Moreover, it allows software engineers to shift perspective, that is, to shift from viewing compliance to ethical principles as a peripheral compliance task and instead treat it as a core software quality attribute, equal in importance to security and performance. In other words, by implementing an ethical framework that treats ethics as a core software quality, engineers can prevent the accumulation of ethical debt and ensure that their software applications are resilient against legal consequences and loss of trust; more importantly, they can ensure their applications do not generate public harm.

References

- ACM. (2018). *ACM code of ethics and professional conduct*. Association for Computing Machinery. <https://www.acm.org/code-of-ethics>
- Carnegie Council for Ethics in International Affairs. (n.d.). Ethical frameworks. <https://carnegiecouncil.org/explore-engage/key-terms/ethical-framework>
- Carnegie Council. (n.d.). *Ethical framework*. Carnegie Council for Ethics in International Affairs. <https://carnegiecouncil.org/explore-engage/key-terms/ethical-framework>
- Floridi, L., & Taddeo, M. (2016). What is data ethics? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2083), 20160360. <https://doi.org/10.1098/rsta.2016.0360>
- GSA. (2020, December). *Federal data strategy: Data ethics framework*. U.S. General Services Administration. <https://resources.data.gov/assets/documents/fds-data-ethics-framework.pdf>
- NIST. (2023, January). Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1). U.S. Department of Commerce. <https://doi.org/10.6028/NIST.AI.100-1>
- OECD. (2019). *Recommendation of the Council on Artificial Intelligence* (OECD/LEGAL/0449). Organization for Economic Co-operation and Development. <https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0449>
- Ricciardi, A. (2026, January 23). Ethical responsibility in data processing. Level Up Coding - Medium. <https://medium.com/p/775ffa165f59>