Carnegie Mellon University
Software Engineering Institute

AI Engineering: 11 Foundational Practices

Recommendations for decision makers from experts in software engineering, cybersecurity, and applied artificial intelligence

Angela Horneman, Andrew Mellinger, and Ipek Ozkaya

THE U.S. DEPARTMENT OF DEFENSE (DOD) is increasingly interested in taking full advantage of the improved capabilities of machine learning (ML) algorithms and building artificial intelligence (AI)-enabled systems that speed up timeliness and accuracy of the decisions made to support DoD missions. With the increased availability of computing resources, applying ML algorithms to models of thousands of parameters—and terabytes and petabytes of data—is now possible.

The availability of ML libraries and off-the-shelf solutions sometimes gives the impression that implementing AI-enabled software systems (AI systems) is easy. However, developing viable and trusted AI systems that are deployed to the field and can be expanded and evolved for decades requires significant planning and ongoing resource commitment. The following practices, informed by our work in software engineering, cybersecurity, and applied AI, are AI engineering recommendations for decision makers. For DoD missions, acquisition and operational perspectives bring additional considerations, which we will address in future reports.

Software Engineering Principles Apply to AI Engineering

Along with the following recommendations, remember that an AI system is a software-intensive system. The established principles of designing and deploying quality software systems that meet their mission goals on time apply to engineering AI systems. Teams should follow modern software and systems engineering practices as well as guidelines such as the Defense Innovation Board's *Ten Commandments of Software*. Teams should strive to deliver functionality on time and with quality, design for architecturally significant requirements (such as security, usability, reliability, performance, and scalability), and plan for sustaining the system for its entire lifetime. However, some concerns in traditional software systems are exaggerated in AI systems, particularly systems that include ML components:

- Engineering teams will need to architect AI systems for inherent uncertainty in their components, data, models, and output.
- Engineering teams will need to adapt to managing different rhythms of change. The rate of change in AI systems is not consistent throughout the system. Data and models may change very frequently, which may or may not imply changes to the rest of the system.
- Managing data will require more resources both up-front and throughout the life of the system.
- Verifying, validating, and securing AI systems will need to account for ambiguity as well as increased attack surface due to frequently changing data and underlying nature of models.

11 Foundational Practices

1. **Ensure you have a problem that both can and *should* be solved by AI.** Start with a well-defined problem, understanding what you want to accomplish and the outcomes you need, while ensuring you have data available to infer those outcomes. Once you know that you have a potential AI problem, verify that other, simpler options—which could be better solutions—do not exist. AI is not a panacea and is often a more complex, less efficient solution for problems where other solutions may already exist.
2. **Include highly integrated subject matter experts, data scientists, and data architects in your software engineering teams.** Effective AI engineering teams consist of experts in the problem domain (subject matter experts), data engineering, model selection and refinement, hardware infrastructure, and software architecting in addition to the other typical software engineering expertise. These team members bring the required skills in algorithm selection, model building, model customization, and data pipeline management that make up the core of AI systems. Include team members who can deal with the sparsity of well-designed tools and the high demands of these systems in terms of performance, scalability, bandwidth, resource management, and versioning.
3. **Take your data seriously to prevent it from consuming your project.** Data ingestion, cleansing, protection, monitoring, and validation are necessary for engineering a successful AI system—and they require tremendous amounts of resources, time, and attention. Ensure that your processes account for
   - changes in the environment
   - possible bias
   - potential for adversarial exploitation throughout the system lifetime

   The output of an AI system is intrinsically tied to the data used to train the system and how well the training data correlates to the problem and the current world. A lot can go wrong with the data, ranging from changes in format that can break an ingest function, to malicious injection of data into a training set that causes an incorrect model or a data leak, to data lacking diversity or sufficient examples of classes of interest. These challenges require a comprehensive data management strategy and oversight function. Automation is critical to managing the data, but teams should balance automation with observability and accessibility.

4. **Choose algorithms based on what you need your model to do, not on their popularity.** Algorithms differ in several important dimensions: what kinds of problems they can solve, how detailed the information in the output is, how interpretable the output and models are, and how robust the algorithm is to adversaries (via manipulating training data, interfering with a feedback loop, and the like). Choose an algorithm that is appropriate for your problem and satisfies your business and engineering needs. As the needs of the system evolve and the environment in which it works changes, the algorithm is likely to change as well.
5. **Secure AI systems by applying highly integrated monitoring and mitigation strategies.** The attack surface of an AI system is expanded due to challenges with understanding how its complex models function and depend on data. These additional attack surface dimensions compound the vulnerability of the traditional hardware and software attack surface. Counteract this circumstance by performing ongoing evaluation and validation—activities that are especially important given present-day conditions of rapid development of attacks and defenses.
6. **Define checkpoints to account for the potential needs of recovery, traceability, and decision justification.** AI systems are acutely sensitive to the dependencies among input data, training data, and models. As such, changes to the version or characteristics of any one can quickly—and sometimes subtly—affect others. In systems where models change periodically, it may be enough to version models with timeframes of use. In systems where models change frequently or continually, carefully consider when and how to correlate input data with the model used to evaluate it, and how to capture and retain that information. Manage these dependencies and versions with care.
7. **Incorporate user experience and interaction to constantly validate and evolve models and architecture.** As much as possible, use an automated approach to capture human feedback on system output and improve (i.e., retrain) models. Monitor user experience to detect issues early, such as degraded performance in the form of system latency or reduced accuracy. Even in low-interaction systems, ensure continued human involvement to monitor for the judgments (practical, ethical, moral, trust, risk related) that computers cannot be coded to evaluate—and for indications of model tampering or system misuse. Be sure to account for user and management automation bias.
8. **Design for the interpretation of the inherent ambiguity in the output.** AI output requires much more interpretation than most other systems. The uncertainty introduced by an AI system might not be acceptable under certain scenarios for the mission and users. Incorporating machine learning components also necessitates designing for output uncertainty and degree of reliability to assist interpreting and assuring the output. Several AI system components may require techniques such as continuous monitoring and instrumentation.
9. **Implement loosely coupled solutions that can be extended or replaced to adapt to ruthless and inevitable data and model changes and algorithm innovations.** The boundaries between the components of an AI system deteriorate more quickly than those in traditional systems due to the entanglement of data. Moreover, the impact of change is heightened due to unanticipated direct and indirect data dependencies. These dependencies may trigger changes in functionality, expected outputs, and even the infrastructure that supports the system. When designing and sustaining AI systems, continuously apply fundamental design principles of engineering to develop loosely coupled, extensible, scalable, and secure systems.
10. **Commit sufficient time and expertise for constant and enduring change over the life of the system.** Teams significantly underestimate resources needed nine out of ten times. Building AI systems requires greater resources initially that need to scale up quickly and significant dedication or resources through the life of the system. These resources include computing, hardware, storage, bandwidth, expertise, and time.
11. **Treat ethics as both a software design consideration and a policy concern.** Evaluate every aspect of the system for potential ethical issues. Account for organizational and societal values in all aspects of the system, from data collection, to decision making, to validation and monitoring of performance and effectiveness. Data collection often raises questions of privacy and in some cases touches other ethical issues, but data collection is not the only area of concern. How the systems will be used (e.g., autonomous military drones), data representation (e.g., ethnic, gender, disability diversity in facial recognition), and model structure (including protected characteristics in credit or employment decisions) can be ethical issues as well.

Final Thoughts

The only constant is change. Designing, deploying, and sustaining AI systems require engineering practices to manage inherent uncertainty in addition to constant and increased rhythm of change. Algorithms, practices, and tools to engineer AI systems are constantly evolving, and changes brought by these systems reach across problem, technology, process, engineering, and cultural boundaries. These AI engineering practices provide a foundation for decision makers to navigate those changes to develop viable, trusted, and extensible systems. As we build and use these systems, we will define better codified engineering and data management practices as well as tools.

About the SEI

The Software Engineering Institute is a federally funded research and development center (FFRDC) that works with defense and government organizations, industry, and academia to advance the state of sthe art in software engineering and cybersecurity to benefit public interest.

Part of Carnegie Mellon University, the SEI is a national resource in pioneering emerging technologies, cybersecurity, software acquisition, and software lifecycle assurance.

Contact Us

CARNEGIE MELLON UNIVERSITY
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE; PITTSBURGH, PA 15213-2612

sei.cmu.edu
412.268.5800 | 888.201.4479
info@sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM19-0624