AI Engineering

Ipek Ozkaya

Technical Director

Engineering Intelligent Software Systems

ozkaya@sei.cmu.edu

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

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.

DM21-0691

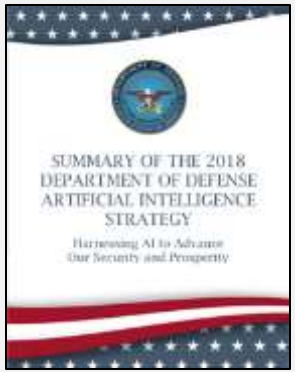
Agenda

AI Engineering and Software Engineering

Foundational AI Practices: Overview and Examples

Misconceptions for AI Systems

AI-enabled systems are software systems!



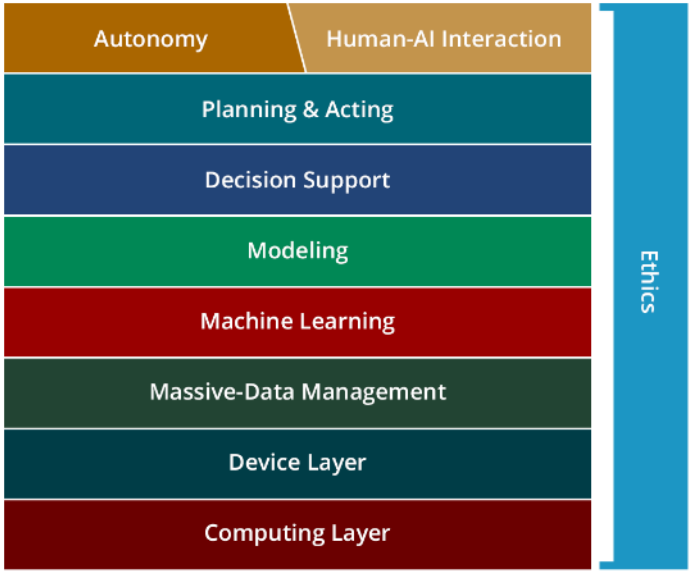
"AI refers to the ability of machines to perform tasks that normally require human intelligence – for example, recognizing patterns, learning from experience, drawing conclusions, making predictions, or taking action – whether digitally or as the smart software behind autonomous physical systems."

Source: *2018 DoD AI Strategy*

An **AI-enabled system** is a **software system** with one or more **AI component(s)** that need to be developed, deployed, and sustained along with the other software and hardware elements of the system.

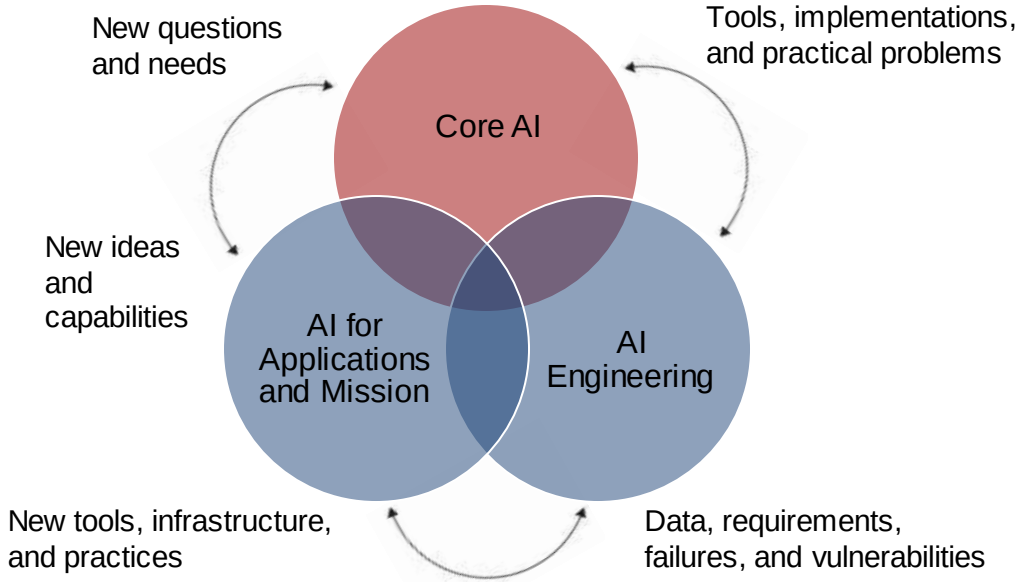
- **Disciplined software engineering** and **cybersecurity practices** are essential starting points in adopting AI.
- The interaction between **software, data, and AI components** (e.g., ML models) requires software design and architecture approaches to be incorporated early and continuously.

AI at CMU and AI at the SEI



CMU AI Stack

A. W. Moore, M. Hebert, S. Shaneman,
"The AI stack: a blueprint for developing and deploying artificial intelligence,"
Proc. SPIE 10635, (4 May 2018); doi: 10.1117/12.2309483



AI at the SEI

SEI Pillars of Work in AI Engineering

AI Engineering is a field of research and practice that combines the principles of systems engineering, software engineering, computer science, and human-centered design to create AI systems in accordance with human needs for mission outcomes.

Human-centered AI

how AI systems are designed to align with humans, their behaviors, and their values

Scalable AI

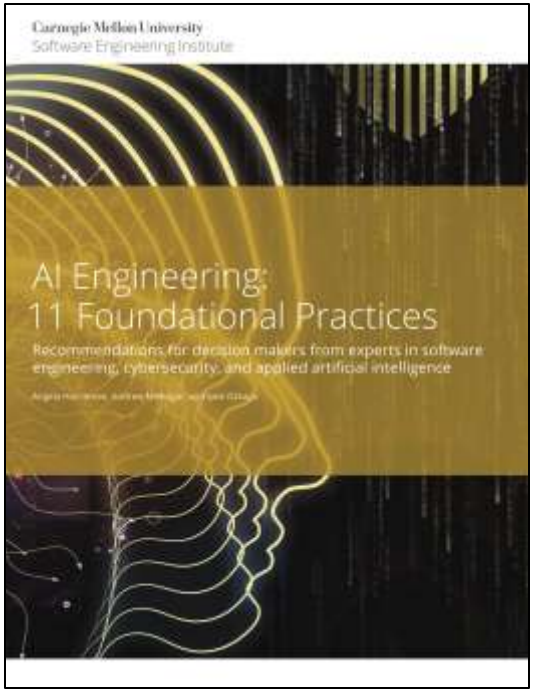
how AI infrastructure, data, and models may be reused across problem domains and deployments.

Robust and Secure AI

how we develop and test resilient AI systems.

https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=735452

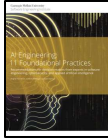
AI Engineering: Foundational Practices



A. Horneman, A. Mellinger, I. Ozkaya. *AI Engineering: 11 Foundational Practices*. Pittsburgh: Carnegie Mellon University Software Engineering Institute, 2019.

Developing, deploying, and sustaining AI-enabled systems requires planning and ongoing resource commitment to take full advantage of the benefits.

Do you have an AI problem?



Ensure that you have a problem that both can and should be solved by AI.

- Your problem might have better, simpler solutions.

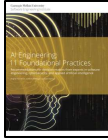
Do you know what measures of success look like for your problem?

- How do you model risk and uncertainty?
- Does your context tolerate and handle uncertainty of results?

If the solution requires data, do you have the necessary data?

- Do you have means to collect, store, and manage data in the long term as well as to secure it and determine who owns it?

Do you have the right team?



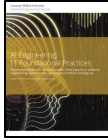
Successful AI system development is a team effort that includes experts in

- software engineering
- data science
- data architecting
- software and system architecting
- continuous integration and deployment
- operations
- the domain

Do you rely on external parties for data?

Are you ready to bring your contractors on board?

Data, data, data, data …



Resources spent in data management WILL consume your project.

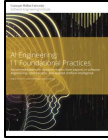
- *Do you have data? Do you have access rights to data? Do you have permission to use the data you have access to? How often does your data change? How will you collect data?....*

Automation and modern infrastructures are critical for successful AI adoption.

Ensure that your data management processes account for

- changes in environment
- adversarial exploitation
- biases in data

Do you understand change?

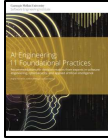


Teams need to plan and design for constant change to manage the “changing anything changes everything” principle*

- Architecting data pipelines to enable incorporating rate of change in data, retraining models, and incorporating new data from training to deployment to maintenance/evolution
- Designing instrumentations to enable traceability of the recommendations to improve explainability
- Changing environment conditions

* Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... & Dennison, D. (2015). Hidden Technical Debt in Machine Learning Systems. In *Proc. 28th Int. Conf. Advances in Neural Information Processing Systems, Vol. 2* (pp. 2503-2511). ACM, 2015.

Do you understand your quality attributes?



Quality attributes are properties of work products or goods by which stakeholders judge their quality.

Software cannot be designed to optimize for all quality attributes.

- Software engineers must select which attributes are most important for the stakeholder needs.
- Stakeholder needs will vary with the context in which a software system is deployed.

The degree to which a software system meets its quality attribute requirements depends on its architecture.

- Architectural decisions are made to promote various quality attributes.

L. Bass, P. Clements, R. Kazman. *Software Architecture Principles and Practices*, 3rd ed. Addison-Wesley, 2012.

Quality attributes drive software architectures

Architecture permits or precludes the *achievement of a system's desired quality attributes*. The strategies for achieving these requirements entail thinking about the structure and behavior of the system.

If you desire…	At a minimum, you need to …
high performance	minimize the frequency and volume of inter-element communication
modifiability	limit interactions between elements
security	manage and protect inter-element communication
availability	determine the properties and behaviors that elements must have and the mechanisms you will employ to address fault detection, fault prevention, and fault recovery
extensibility	limit interactions between elements, isolate data types, and abstract common services

High-Priority Quality Attributes for AI-Enabled Systems

AI engineering software design challenge	You will desire…	At a minimum, you need to …
AI introduces new attack surfaces.	security	• decouple model changes from the rest of the system • build in capabilities to modify the systems to ease deploying retrained models
Tight coupling of data and models may limit implementing privacy protections.	privacy	• decouple data stores and their interactions with other systems as much as possible • isolate changes and updates to as few locations as possible
Software update cycles may not adequately address data changes and their impact over time.	data centricity	• ensure that uncertainty, availability, and scalability of data are key architecture drivers for system design • implement monitoring components to observe and manage data changes over time
Output of AI components is not human interpretable.	explainability	• decouple model changes from changes to the rest of the system • introduce observability mechanisms into the system
Rate of change that impacts software and AI components can vary significantly.	sustainability	• express rate of change as an architectural concern • build in monitoring components for both the system and the AI components

L. Pons, I. Ozkaya. Priority Quality Attributes for Engineering AI-enabled Systems. *Association for the Advancement of Artificial Intelligence AI in Public Sector Workshop.* Washington, DC, November 7-9, 2019.

AI Exacerbates Existing SE Challenges

Specifying systems:

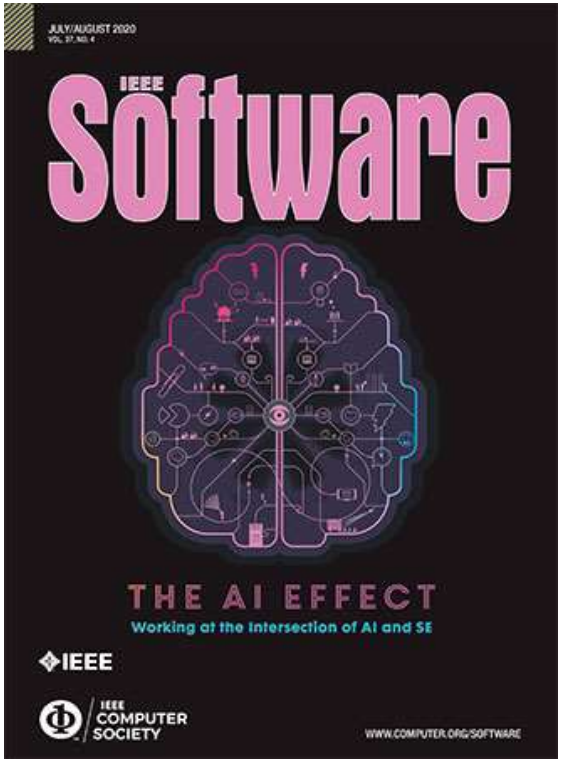
- The level of specification of AI systems depends on the level of uncertainty in the discovery process. Sometimes uncertainty is low to none.

Avoiding hidden dependencies:

- Understanding data and shared resource dependencies is not only an AI system problem, but also a software system problem.

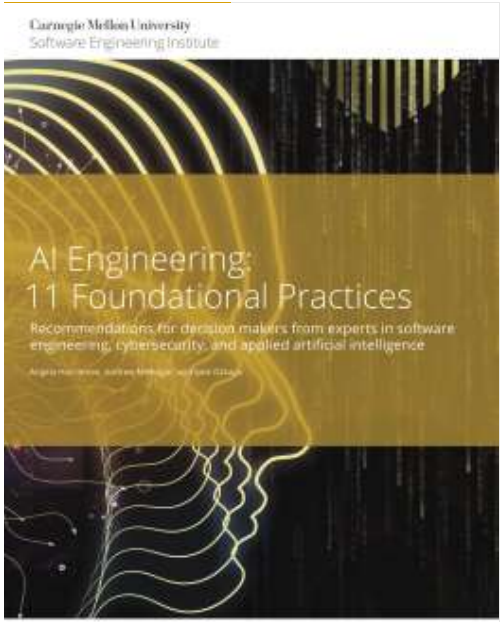
Relying too much on frameworks and tools:

- Existing frameworks, model libraries, tools, and deployment environments help, but do not replace designing for scalability, observability, and sustainability of AI systems.

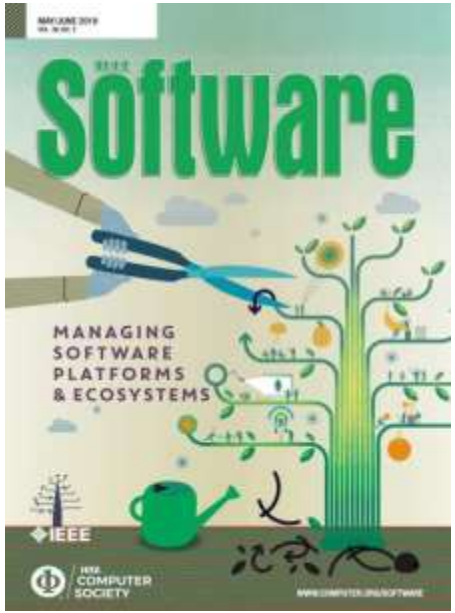


Ipek Ozkaya. *What Is Really Different in Engineering AI-Enabled Systems? IEEE Softw. 37(4): 3-6 (2020).*

Recommendations



A. Horneman, A. Mellinger, I. Ozkaya. *AI Engineering: 11 Foundational Practices, Sept.* 2019



I. Ozkaya. *Ethics Is a Software Design Concern.* IEEE Softw. 36(3): 4-8 (2019).

Ensure that you have a problem that both can and *should* be solved by AI.

Take your data seriously to prevent it from consuming your project.

Incorporate user experience and interaction to constantly validate and evolve models and architecture.

Treat ethics as both a software design consideration and a policy concern.

Contact Information

Ipek Ozkaya

Technical Director

Engineering Intelligent Software Systems

Software Engineering Institute

Carnegie Mellon University

ozkaya@sei.cmu.edu