# Accelerating Software Acquisition Using Generative AI for Regulatory Compliance

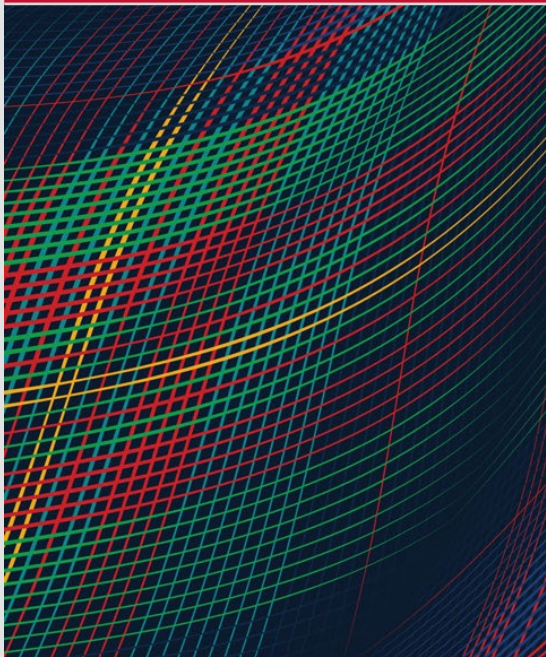
**MAY 7, 2025**

John E. Robert, Deputy Director, Software Solutions Division, jer@sei.cmu.edu
Carlos Olea, SEI Visiting Scientist, colea@sei.cmu.edu
Yash Hindka, SEI Member of Technical Staff, yhindka@sei.cmu.edu
Nanette Brown, Senior SEI Member of Technical Staff, nb@sei.cmu.edu
Douglas C. Schmidt, Dean of Computing, Data Sciences & Physics at William & Mary, douglas.c.schmidt@wm.edu





Carnegie Mellon University
Software Engineering Institute



# Document Markings



The following markings MUST be included in work product when attached to this form and when it is published.

For purposes of double anonymous peer review, markings may be temporarily omitted to ensure anonymity of the author(s).



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.



DM25-0657

# Objectives

## Motivation for AI-Augmented Acquisition

## Software Acquisition Using Generative AI for Regulatory Compliance

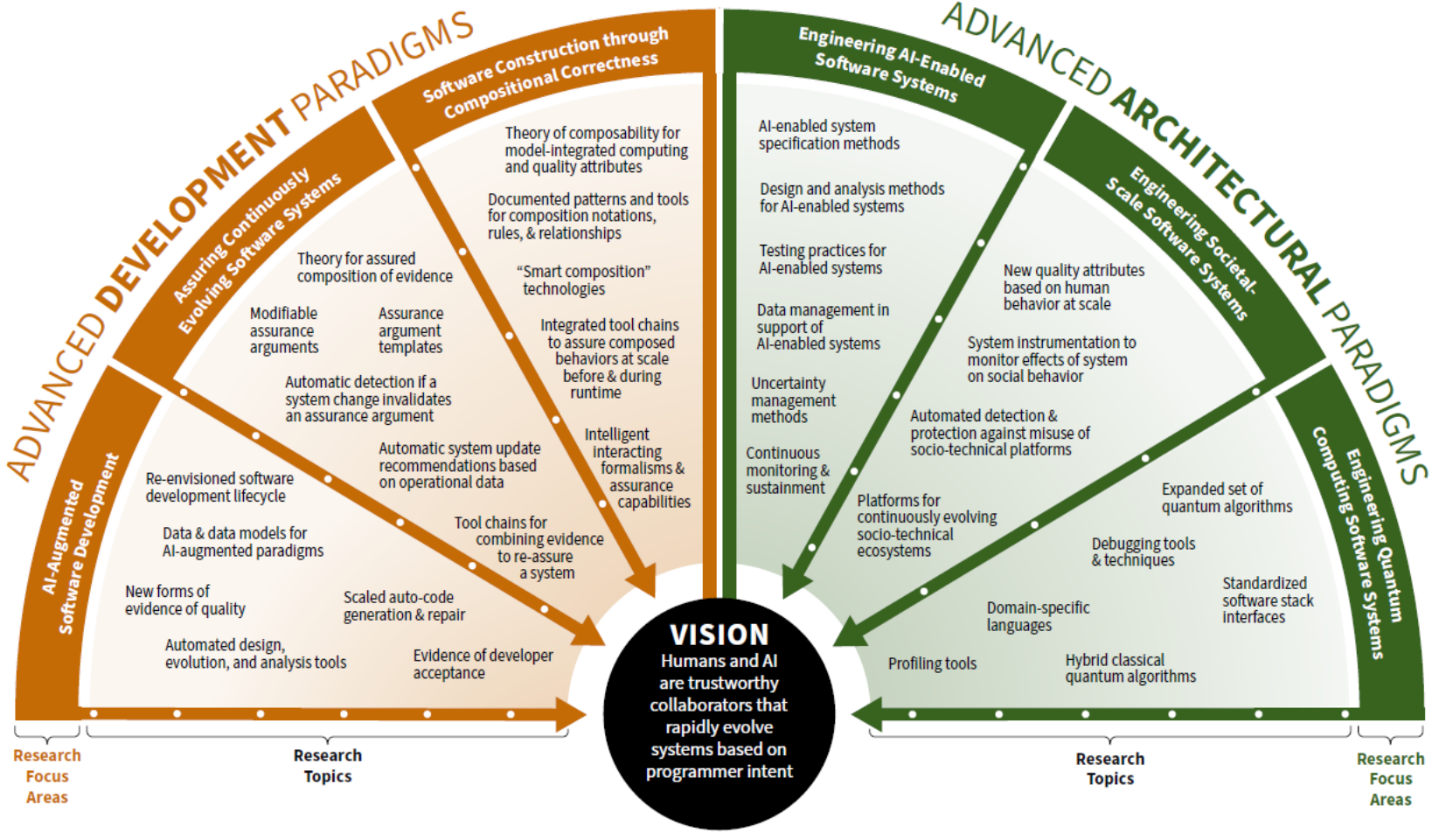

Carnegie Mellon University
Software Engineering Institute

The CMU SEI-led study *Architecting the Future of Software Engineering: A National Agenda for Software Engineering Research & Development* laid out a multi-year roadmap for engineering next-generation software-reliant systems.

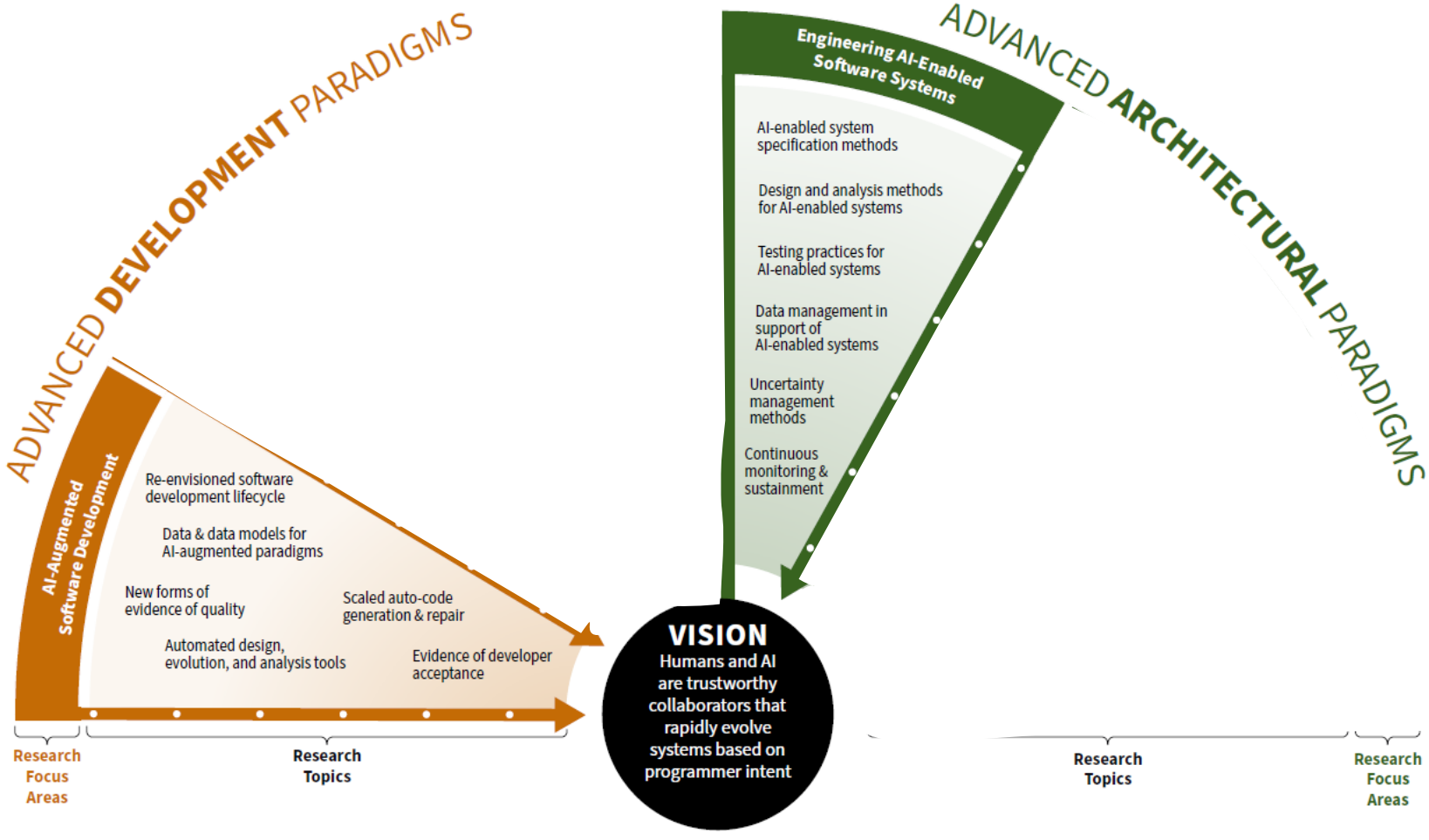
Study available online at https://www.sei.cmu.edu/go/national-agenda

# Software Engineering Research Roadmap (10-15 Year Horizon)

Carnegie Mellon University
Software Engineering Institute



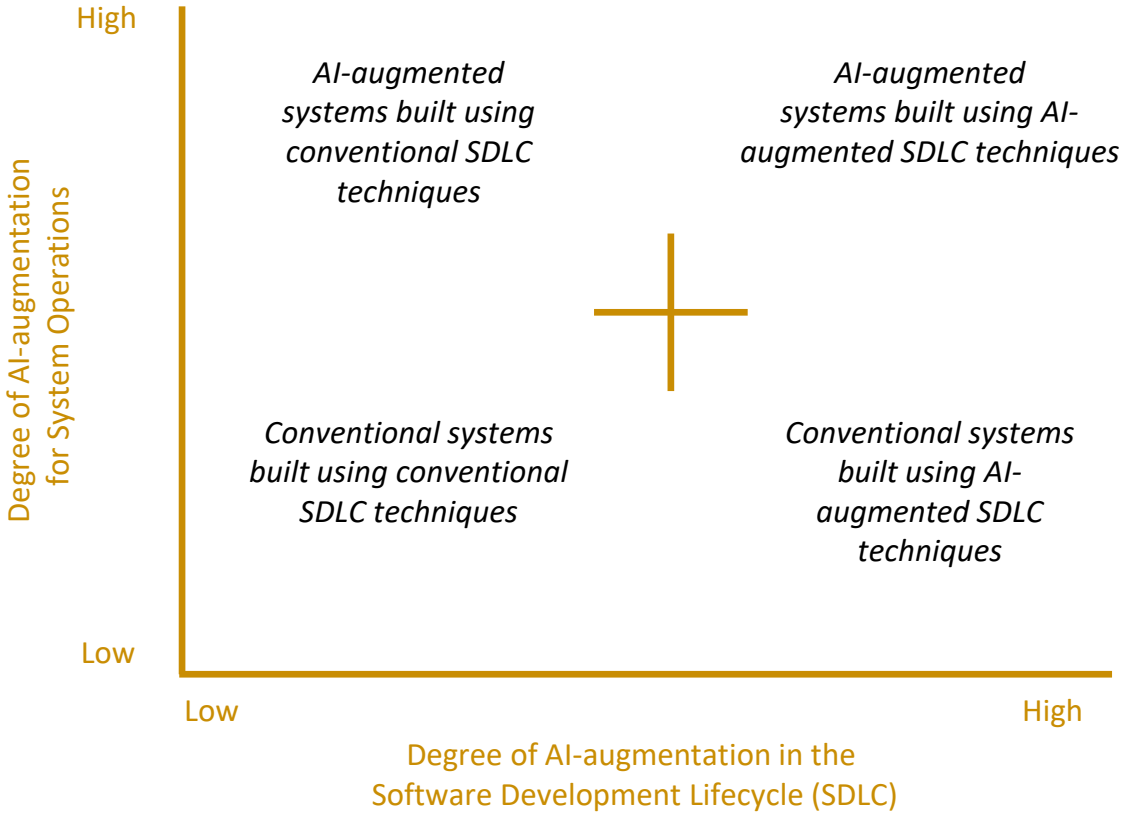
# Software Engineering Research Roadmap (10-15 Year Horizon)



# How Advances in Generative AI Affected Our Study Findings

- Based on extensive experience, we've refined our taxonomy to focus on the degree of AI-augmentation for the software development lifecycle (SDLC) & system operations

Carnegie Mellon University Software Engineering Institute




## Application of Large Language Models (LLMs) in Software Engineering: Overblown Hype or Disruptive Change?

IPEK OZKAYA, ANITA CARLETON, JOHN E. ROBERT, AND DOUGLAS SCHMIDT (VANDERBILT UNIVERSITY)

OCTOBER 2, 2023

Has the day finally arrived when large language models (LLMs) turn us all into better software engineers? Or are LLMs creating more hype than functionality for software development, and, at the same time, plunging everyone into a world where it is hard to distinguish the perfectly formed, yet sometimes fake and incorrect, code generated by artificial intelligence (AI) programs from verified and well-tested systems?

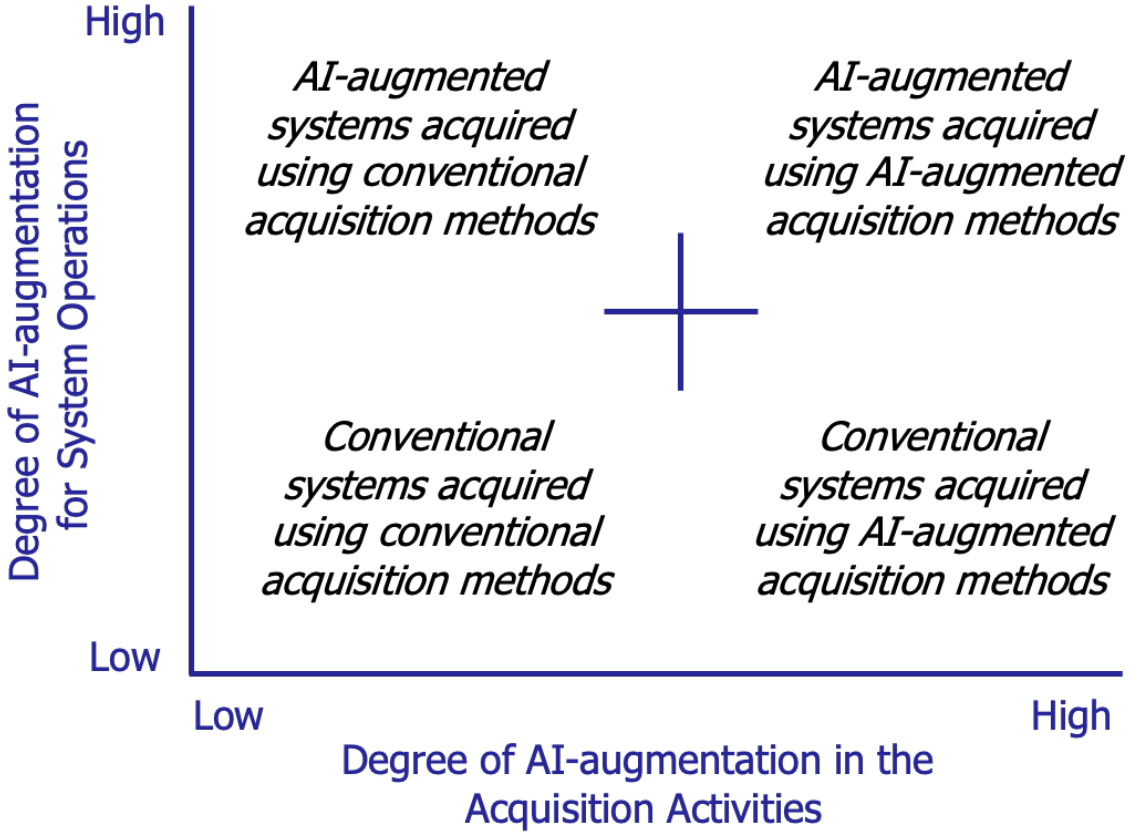
### LLMs and Their Potential Impact on the Future of Software Engineering

This blog post, which builds on ideas introduced in the IEEE paper *Application of Large Language Models to Software Engineering Tasks: Opportunities, Risks, and Implications* by Ipek Ozkaya, focuses on opportunities and cautions for LLMs in software development, the implications of incorporating LLMs into software-reliant systems, and the areas where more research and innovations are needed to advance their use in software engineering. The reaction of the software engineering community to the accelerated advances that LLMs have demonstrated since the final quarter of 2022 has ranged from snake oil to no help for programmers to the end of programming and computer science education as we know it to revolutionizing the software development process. As is often the case, the truth lies somewhere in the middle, including new opportunities and risks for developers using LLMs.

Research agendas have anticipated that the future of software engineering would include an AI-augmented software development lifecycle (SDLC), where both software engineers and AI-enabled tools share roles, such as copilot, student, expert, and supervisor. For example, our November 2021 book *Architecting the Future of Software Engineering: A National Agenda for Software Engineering Research and Development* describes a research path toward humans and AI-enabled tools working as trusted collaborators. However, at that time (a year before ChatGPT was released to the public), we didn't expect these opportunities for collaboration to emerge so rapidly. The figure below, therefore, expands upon the vision presented in our 2021 book to codify the degree to which AI augmentation can be applied in both system operations and the software development lifecycle (Figure 1), ranging from conventional methods to fully AI-augmented methods.

See application-of-large language-models-llms-in-software-engineering-overblown-hype-or-disruptive-change

# Impact of AI on Software Acquisition

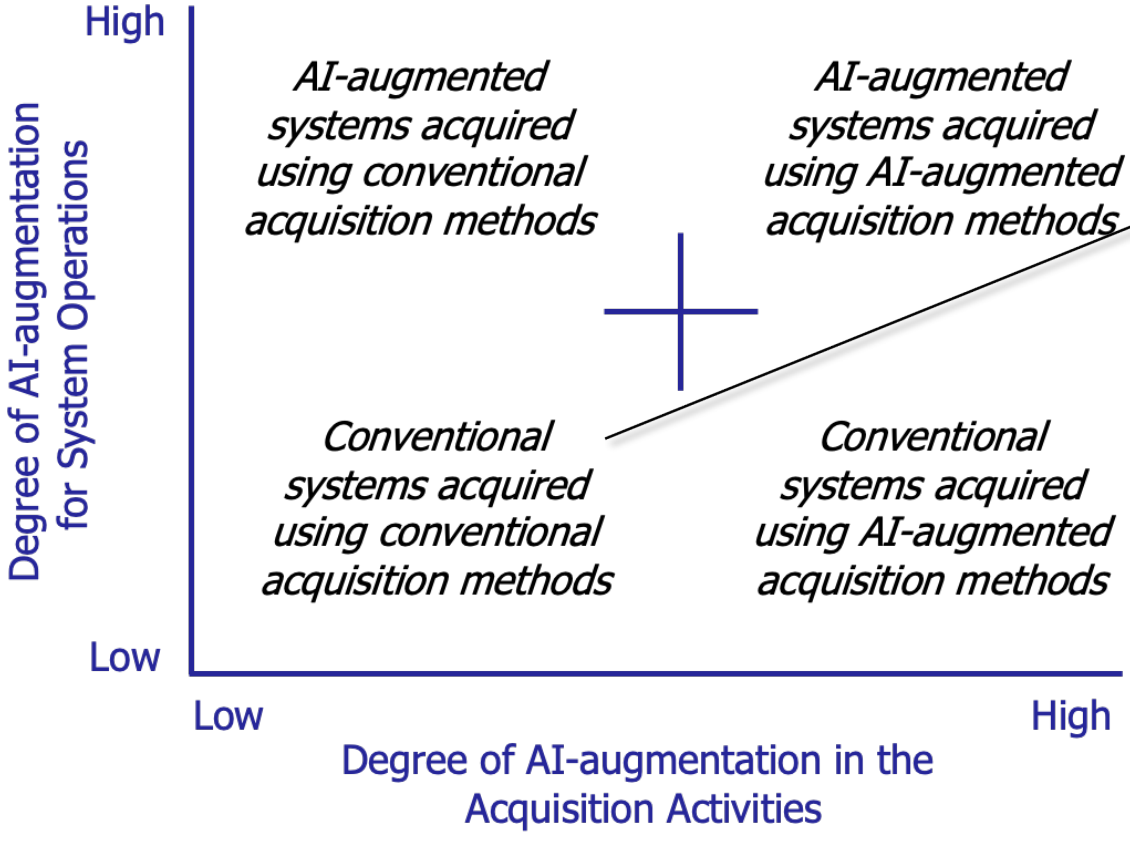




*J. Robert, J. Ivers, D. C. Schmidt, I. Ozkaya, & S. Zhang, "The Future of Software Engineering & Acquisition with Generative AI," Crosstalk, June 2024.*

# Impact of AI on Software Acquisition

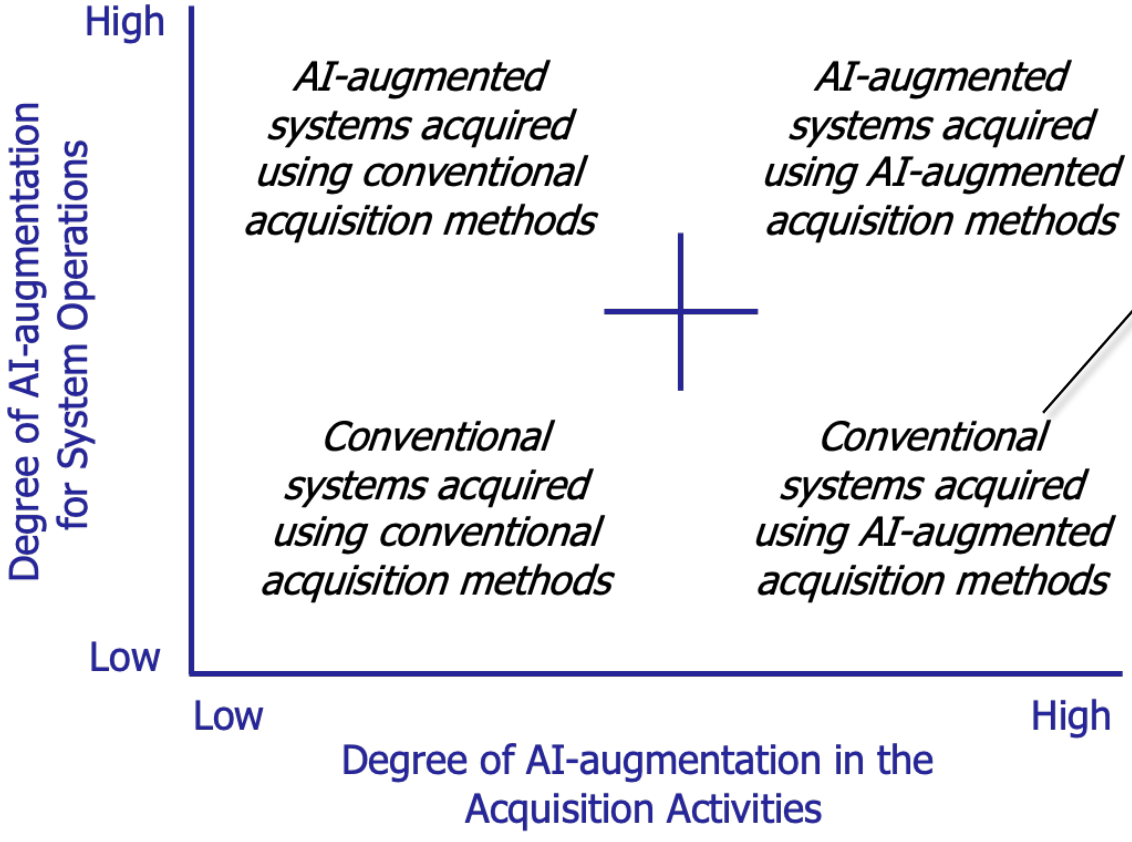





*A military-grade GPS satellite system using traditional data transmission & encryption for operations & is developed using conventional acquisition processes without any AI-augmented tools or methods.*

*J. Robert, J. Ivers, D. C. Schmidt, I. Ozkaya, & S. Zhang, "The Future of Software Engineering & Acquisition with Generative AI," Crosstalk, June 2024.*

# Impact of AI on Software Acquisition

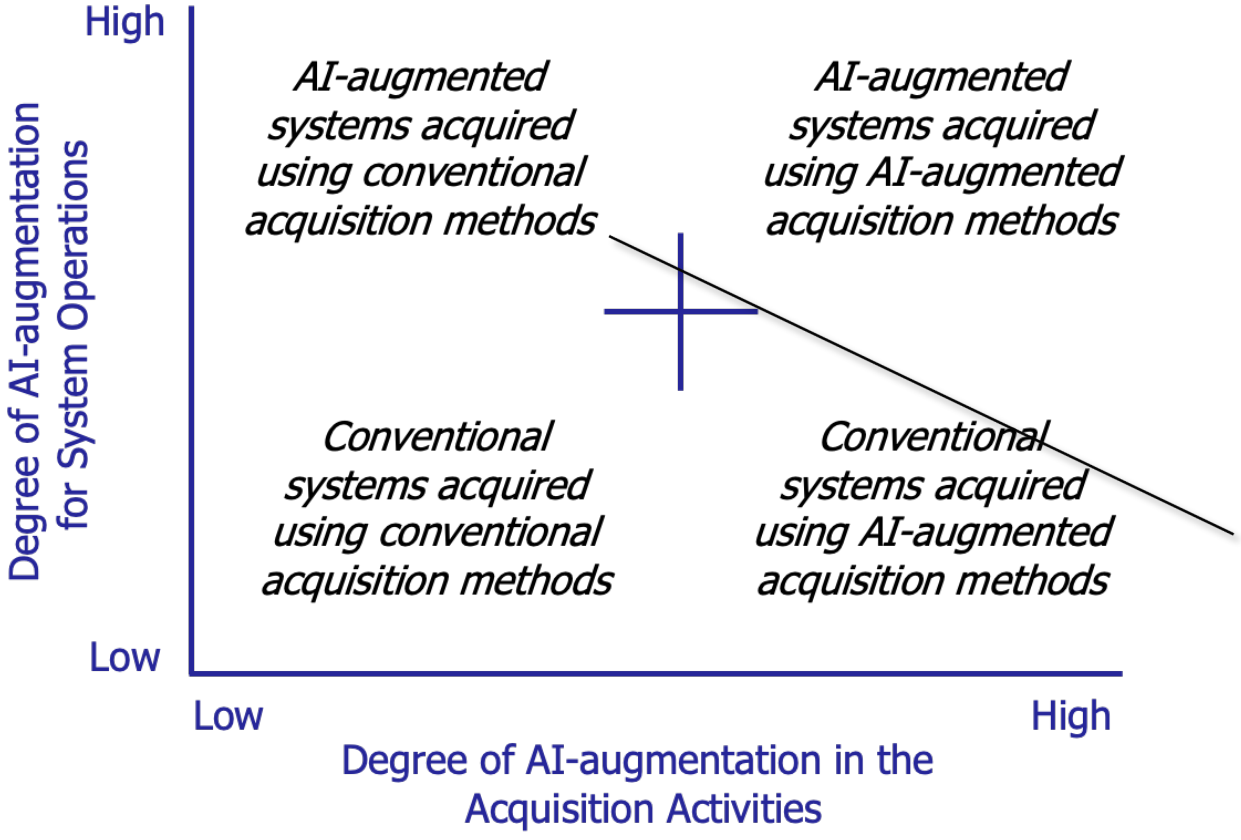



*GPS-guided munition where the content is not AI-augmented, but the acquisition activities employ AI-assistance in identifying & analyzing relevant regulations, standards, & potential security risks.*

*J. Robert, J. Ivers, D. C. Schmidt, I. Ozkaya, & S. Zhang, "The Future of Software Engineering & Acquisition with Generative AI," Crosstalk, June 2024.*

# Impact of AI on Software Acquisition





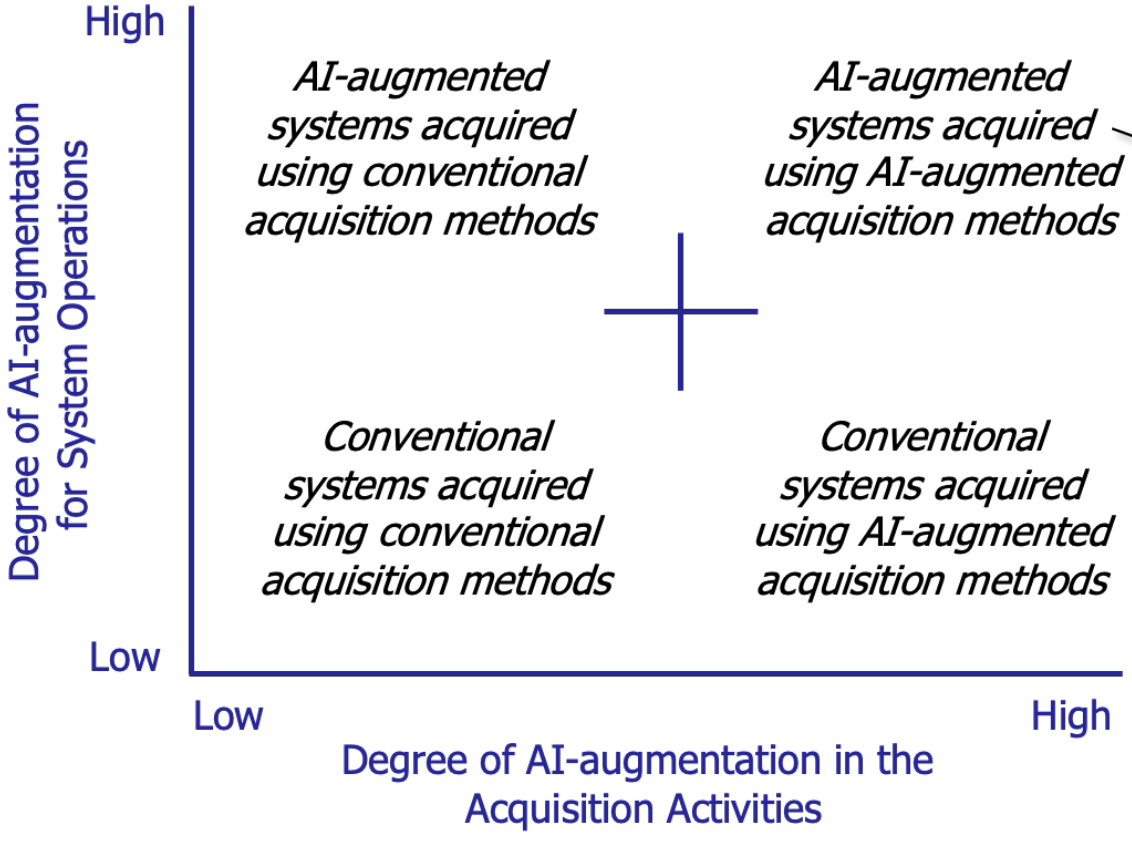


*A radar system that employs machine learning to identify & prioritize possible targets, but the system is acquired using conventional methods*

*J. Robert, J. Ivers, D. C. Schmidt, I. Ozkaya, & S. Zhang, "The Future of Software Engineering & Acquisition with Generative AI," Crosstalk, June 2024.*

# Impact of AI on Software Acquisition

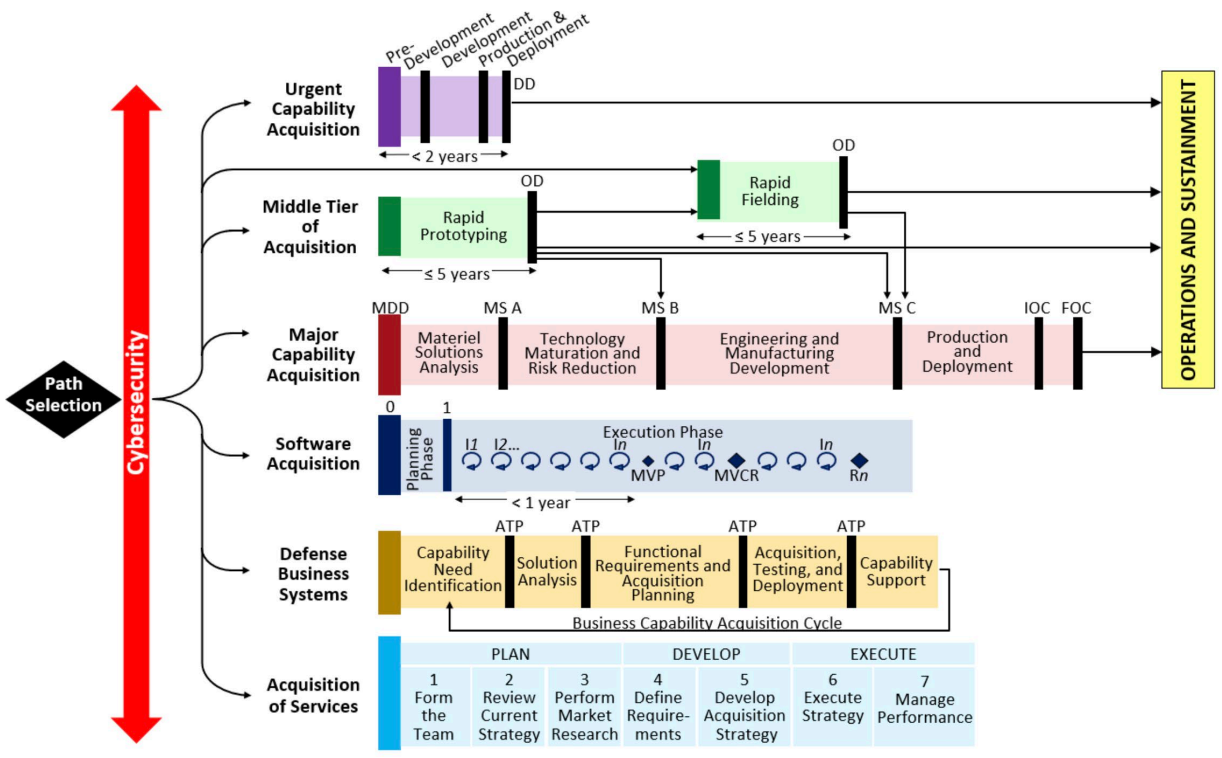





*An autonomous vehicle or platform that employs AI to navigate while also using AI-augmented acquisition processes, methods, & tools, such as text summarization & semi-automated regulatory compliance.*


*J. Robert, J. Ivers, D. C. Schmidt, I. Ozkaya, & S. Zhang, "The Future of Software Engineering & Acquisition with Generative AI," Crosstalk, June 2024.*

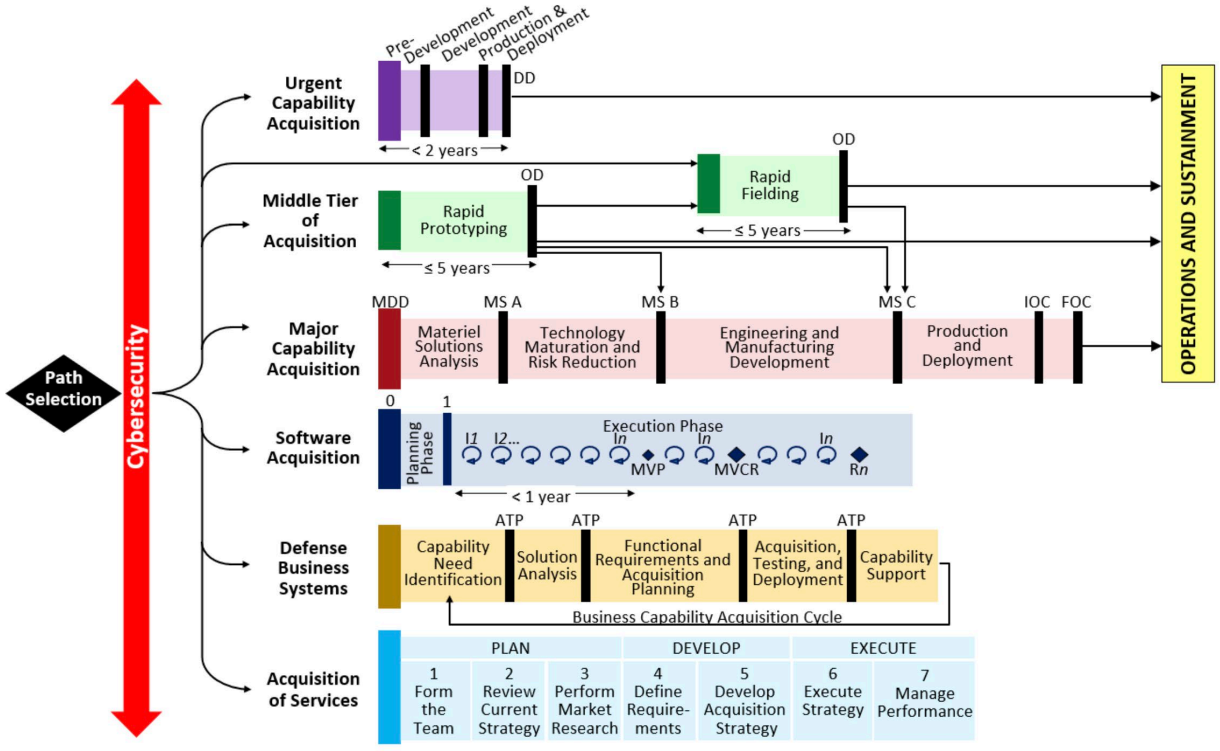
# DOD Software Acquisition Use Cases




As software-reliant systems become increasingly essential to the success of DoD missions there has been greater emphasis on codifying software acquisition strategies  policies

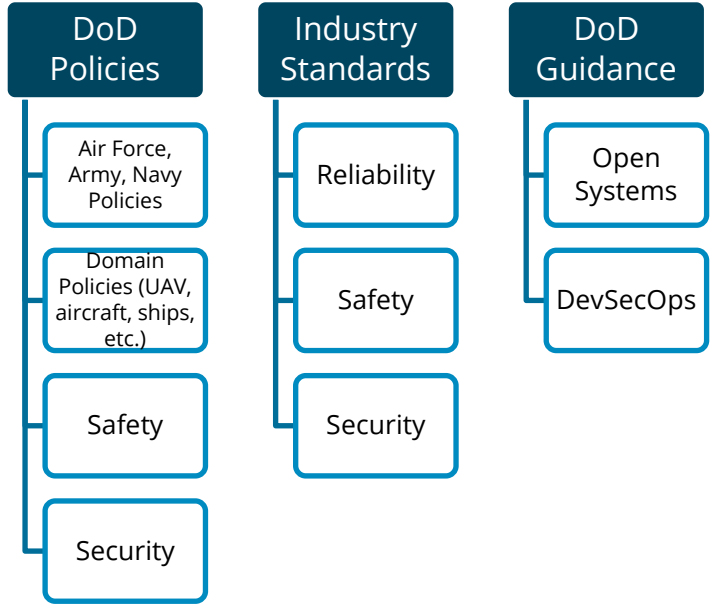
https://aaf.dau.edu/aaf/aaf-pathways/

# DOD Software Acquisition Use Cases




## Challenges for acquisition professionals

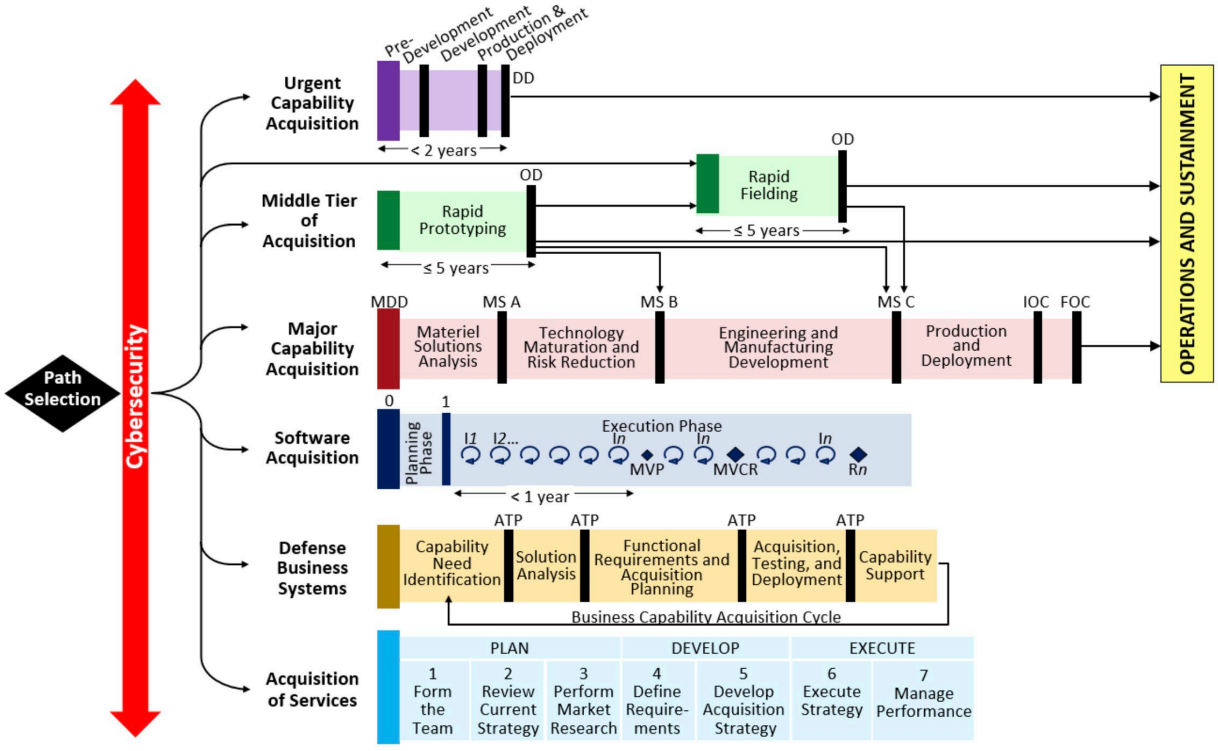
- Heavily regulated environment with multiple levels of policy & legal responsibilities




https://aaf.dau.edu/aaf/aaf-pathways/

# DOD Software Acquisition Use Cases

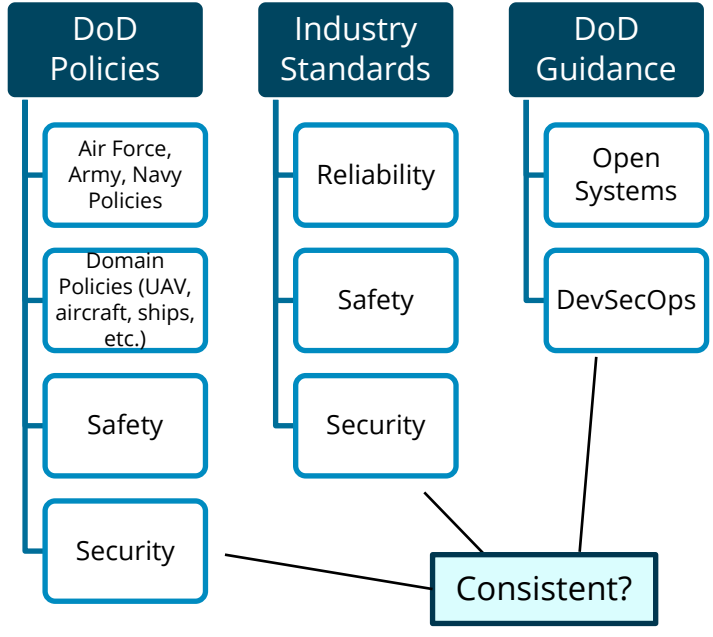




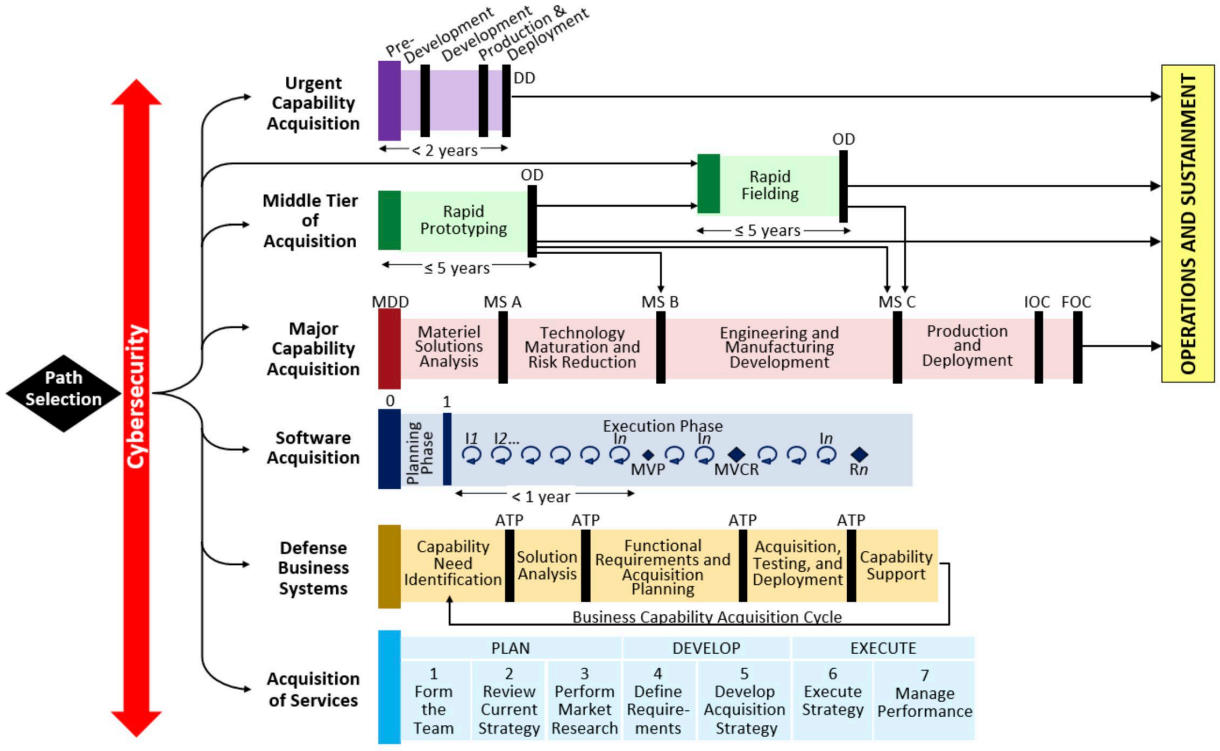

https://aaf.dau.edu/aaf/aaf-pathways/

## Challenges for acquisition professionals

- Heavily regulated environment with multiple levels of policy & legal responsibilities



# DOD Software Acquisition Use Cases




https://aaf.dau.edu/aaf/aaf-pathways/

## Challenges for acquisition professionals

- Heavily regulated environment with multiple levels of policy & legal responsibilities
- Volume of policy, guidance, & standards documents
- Concurrency of activities within one program

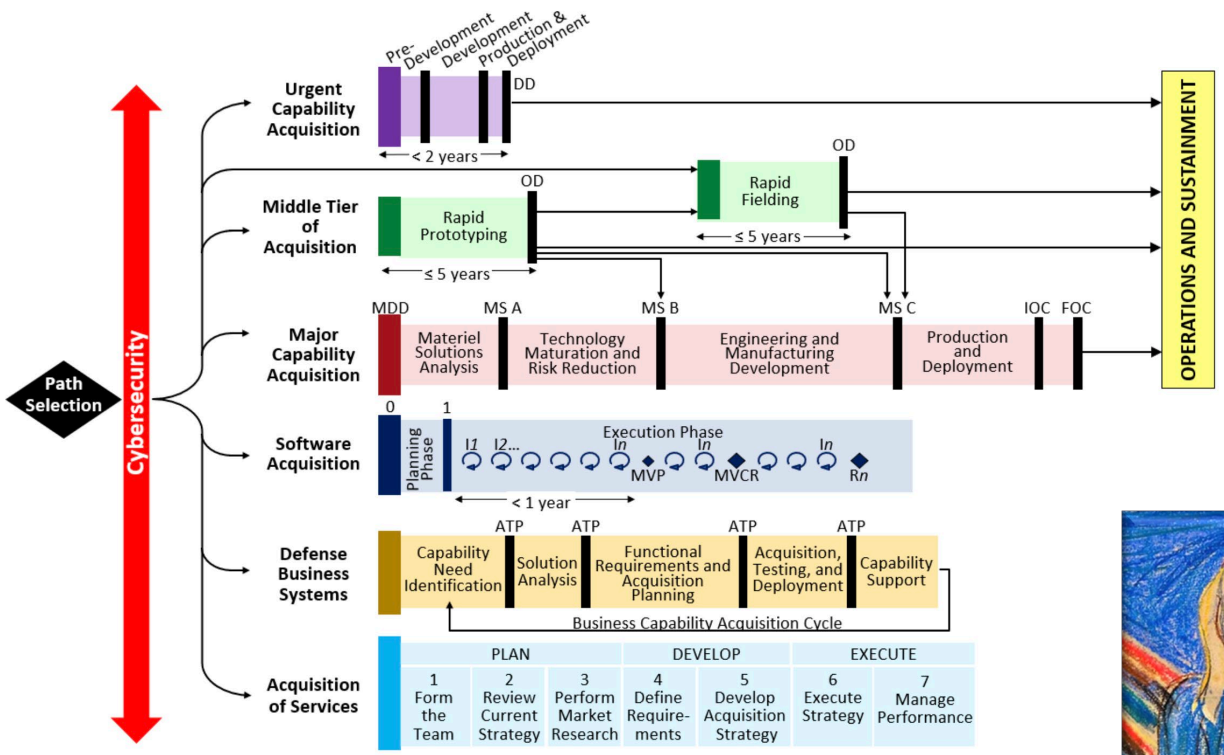
**Release 1**
- Development
- Testing

**Release 2**
- Design
- Development

**Release 3**
- Requirements

# DOD Software Acquisition Use Cases

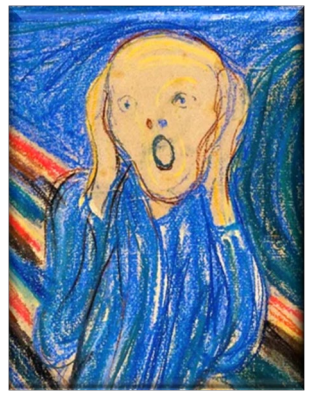


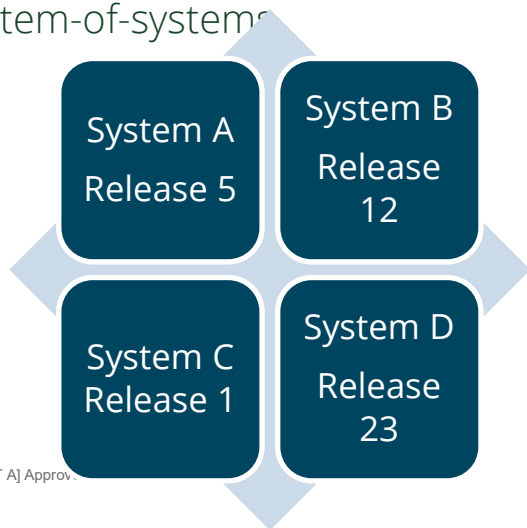



https://aaf.dau.edu/aaf/aaf-pathways/

## Challenges for acquisition professionals

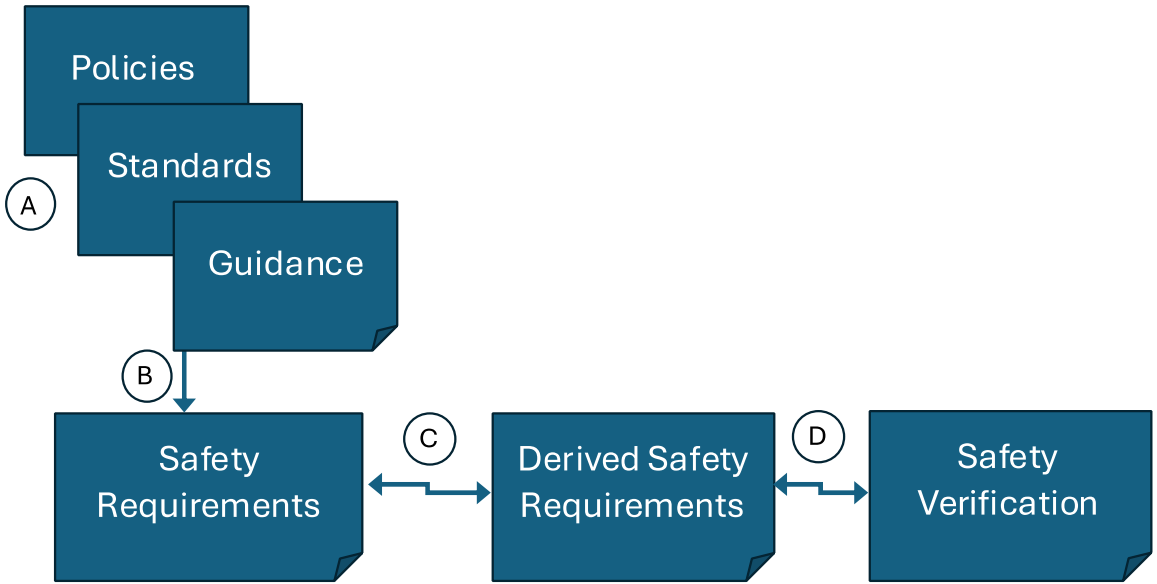
- Heavily regulated environment with multiple levels of policy & legal responsibilities
- Volume of policy, guidance, & standards documents
- Concurrency of activities within one program & across multiple programs in a system-of-systems





# Acquisition Example: Software Safety





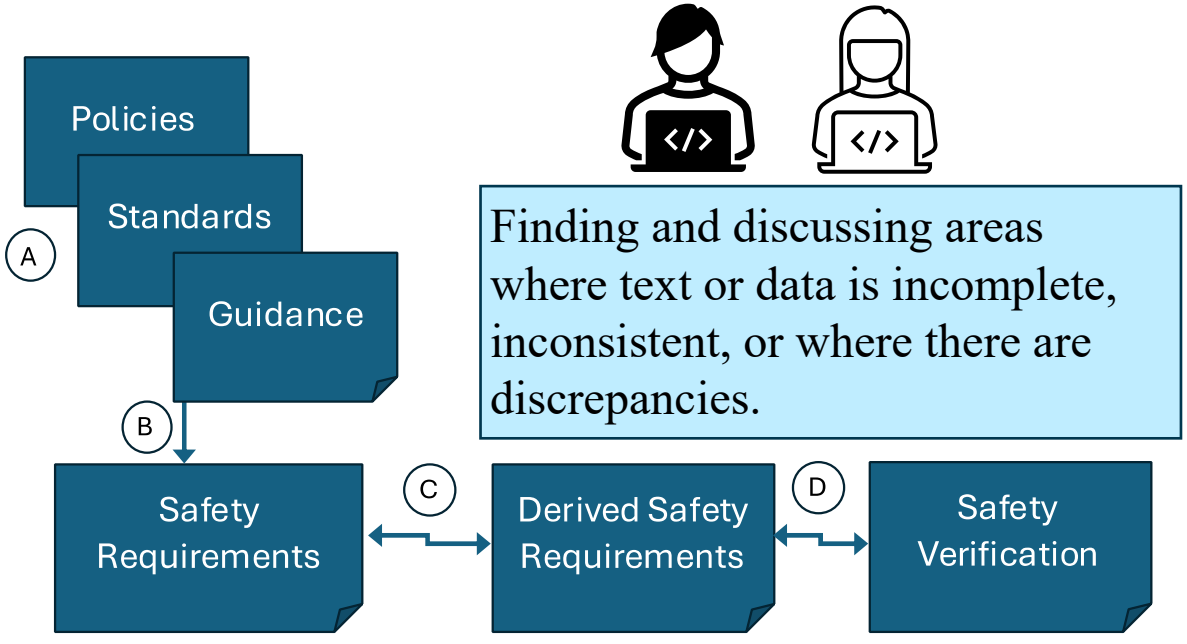

A. Regulatory documents – Policies, standards, and guidance documents.

B. Safety Requirements –Document that lists safety requirements derived from the regulatory documents that are relevant to a specific system.

C. Derived Safety Requirements – Document that captures derived safety requirements that decomposes the safety requirements from top system level to every software module or component in the system.

D. Safety Verification – Collection of safety verification documents and data, that summarize the verification evidence (test results, analysis results, etc.) when compared to the derived safety requirements.

# Acquisition Example: Software Safety



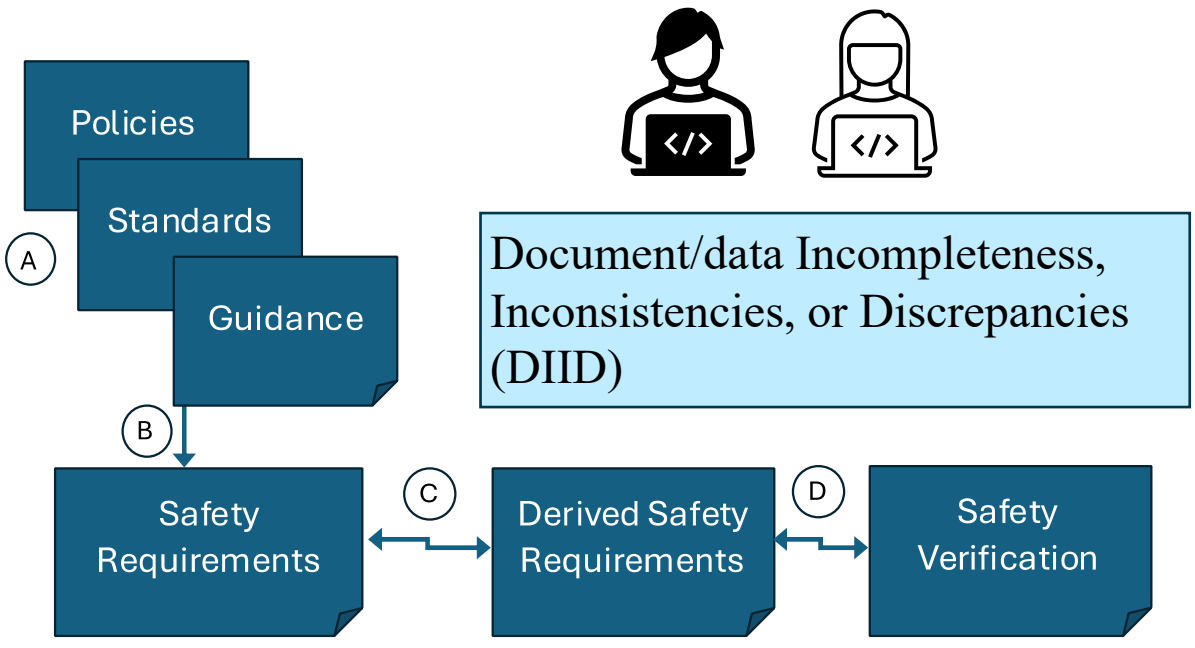

A. Regulatory documents – Policies, standards, and guidance documents.

B. Safety Requirements –Document that lists safety requirements derived from the regulatory documents that are relevant to a specific system.

C. Derived Safety Requirements – Document that captures derived safety requirements that decomposes the safety requirements from top system level to every software module or component in the system.

D. Safety Verification – Collection of safety verification documents and data, that summarize the verification evidence (test results, analysis results, etc.) when compared to the derived safety requirements.

# Acquisition Example: Software Safety




A. Regulatory documents – DIID can occur between multiple policy, standards, and guidance documents.

B. Safety Requirements – DIID can occur between regulatory documents and safety requirements document that lists safety requirements derived from the regulatory documents that are relevant to a specific system.

C. Derived Safety Requirements – DIID can occur between safety requirements documents and derived safety requirements document that extends the safety requirements from top level to every software module or component in the system.

D. Safety Verification – DIID can occur with safety verification documents that summarize the verification evidence (test results, analysis results, etc.) when compared to the derived safety requirements.

# AI-Augmented Acquisition



## Challenges for acquisition professionals

- Heavily regulated environment with multiple levels of policy & legal responsibilities
- Volume of policy, guidance, & standards documents
- Concurrency of activities within one program & across multiple programs in a system-of-systems

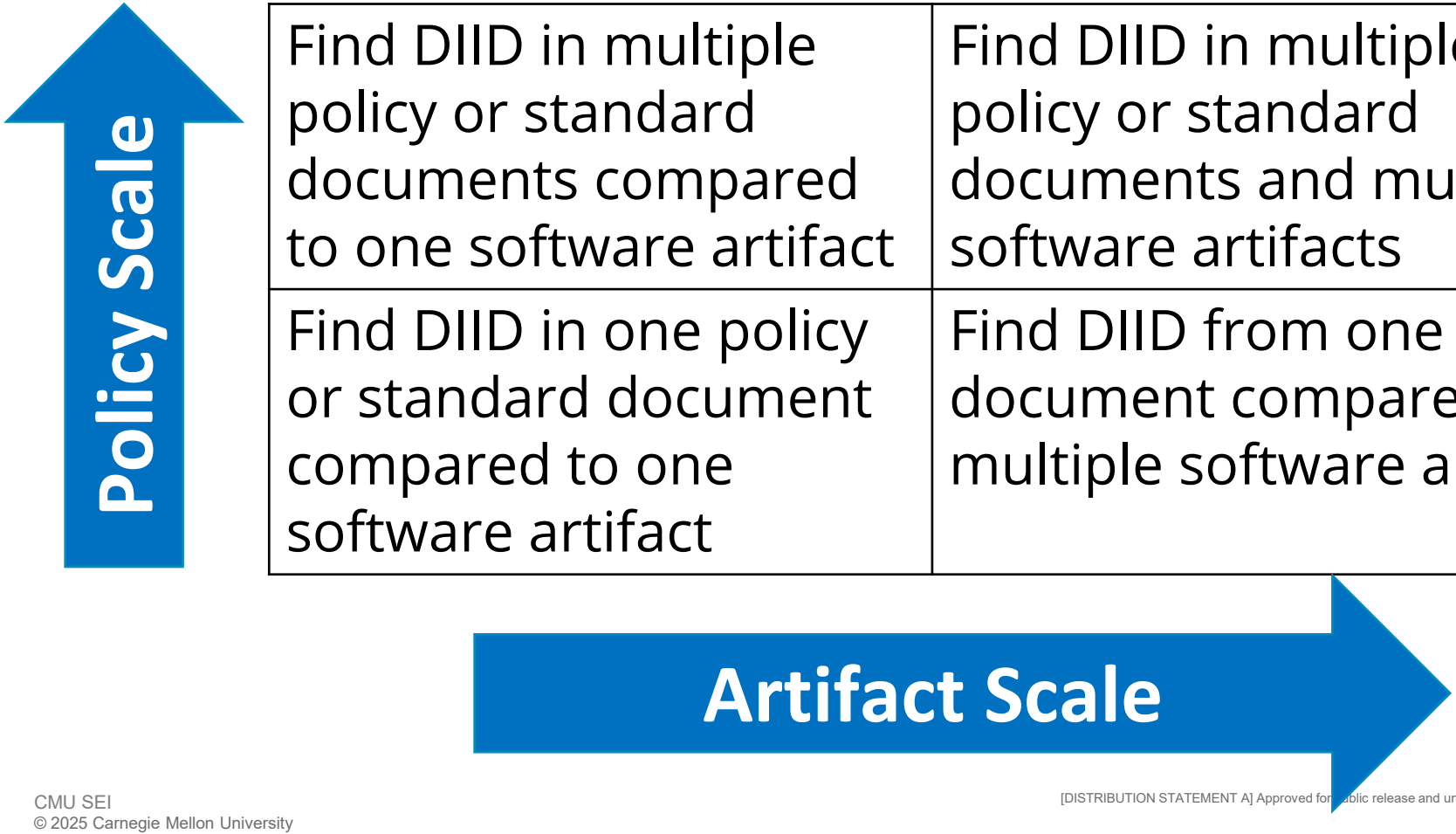

## AI-Augmented Acquisition

- AI tools to augment people
- Intelligent automation to scale

*An overarching challenge for software acquisition professionals is detecting & remediating incompleteness, inconsistencies, & discrepancies within/between documents and software engineering artifacts*

# Dimensions of Finding DIID in Acquisition





# Software Acquisition Using Generative AI for Regulatory Compliance

Carnegie Mellon University
Software Engineering Institute

# Role of DIID in Regulatory Risk and Non-Compliance

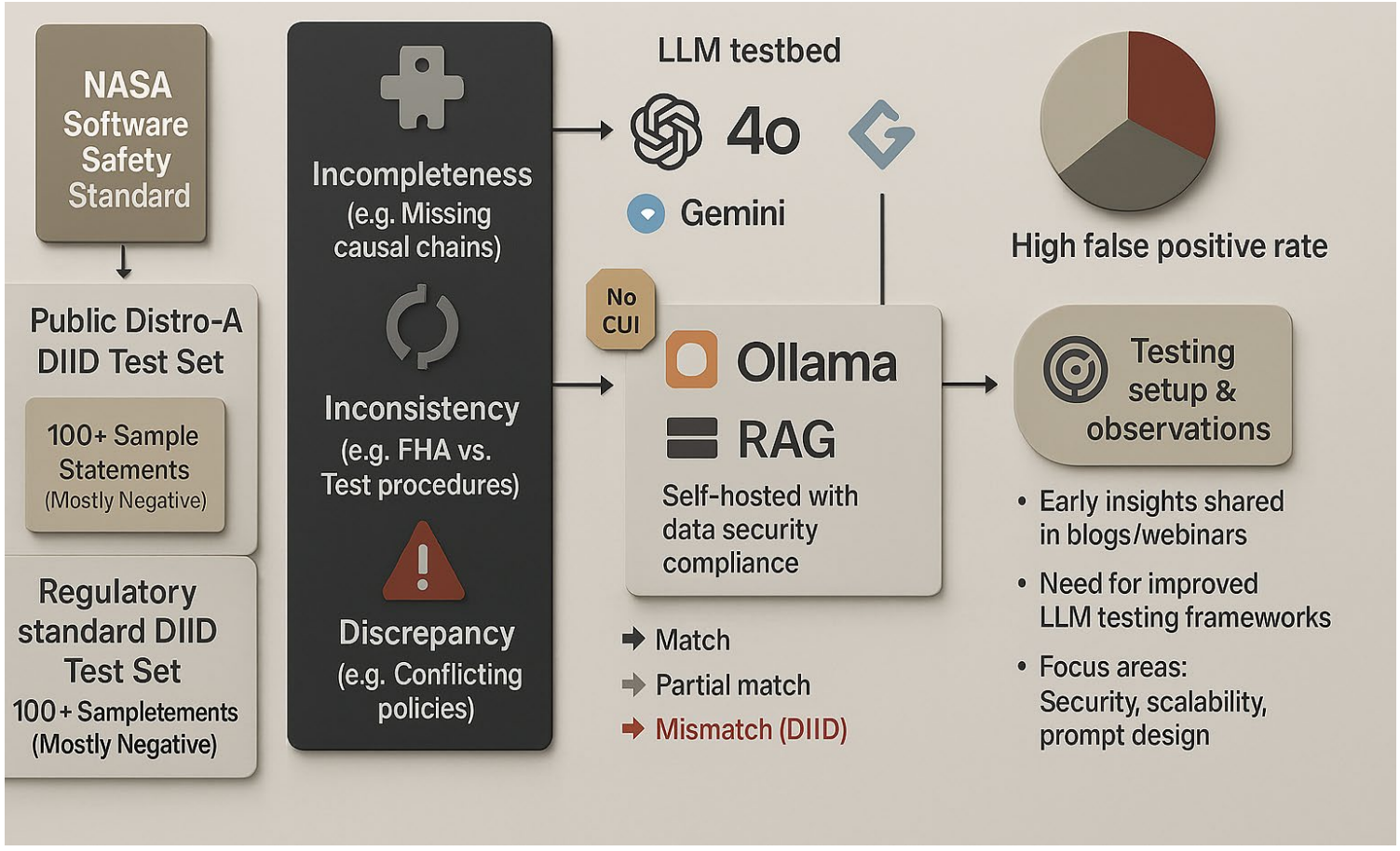
# Types of DIID



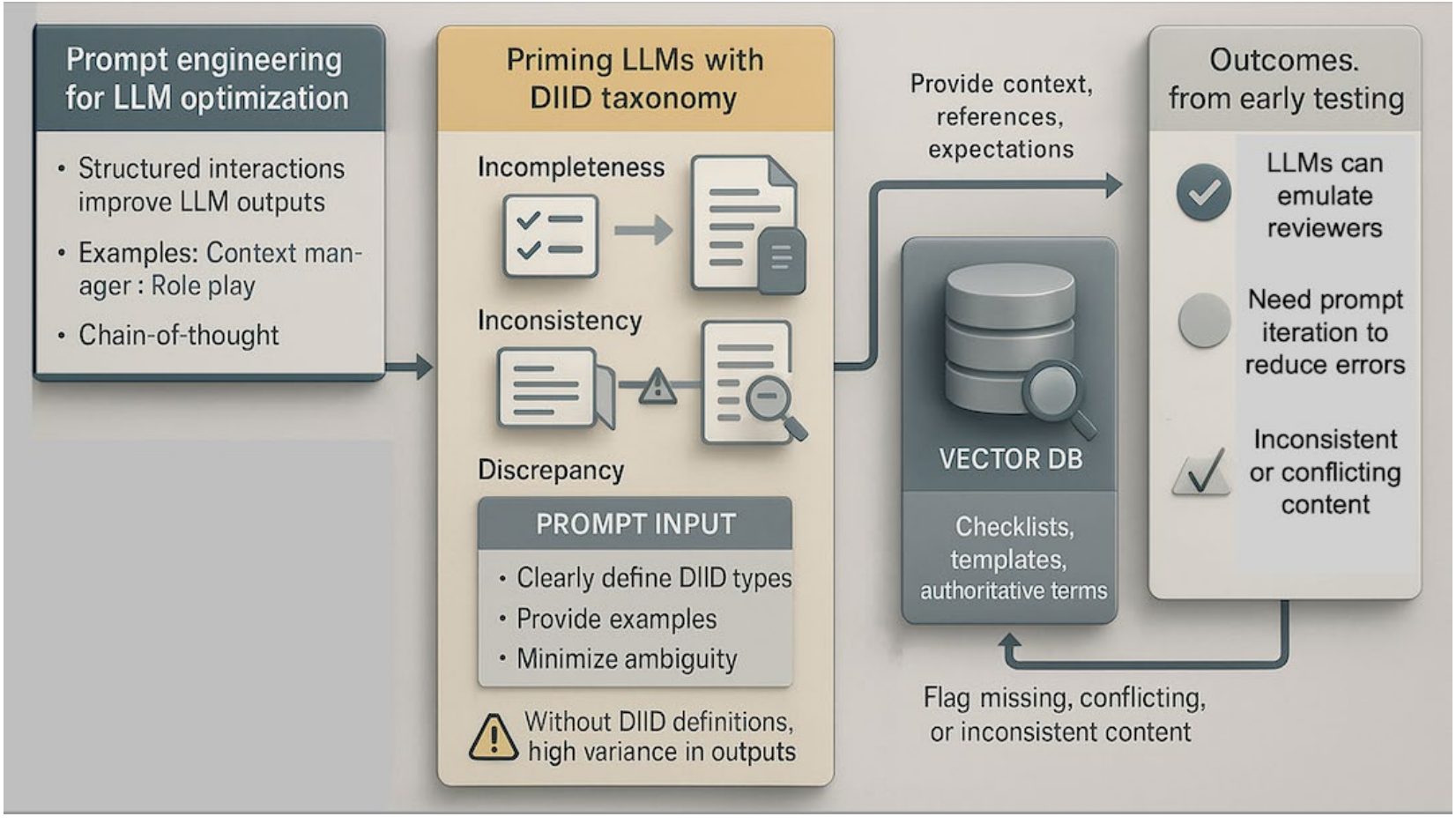
| Incompleteness | Inconsistency | Discrepancy |
|---|---|---|
| • Incomplete: Important context or terms are missing. | • Terminology: Using different terms interchangeably without clear definitions or consistency.<br>• Structural: Lack of uniform structure in presenting information. | • Factual discrepancies: Conflicting factual information.<br>• Policy or procedural discrepancies: Deviations from established protocols.<br>• Narrative discrepancies: Different user stories fail to align.<br>• Theoretical discrepancies: Actual results conflict with theoretical predictions. |

# AI-Augmented DIID Detection in DoD Safety Standards

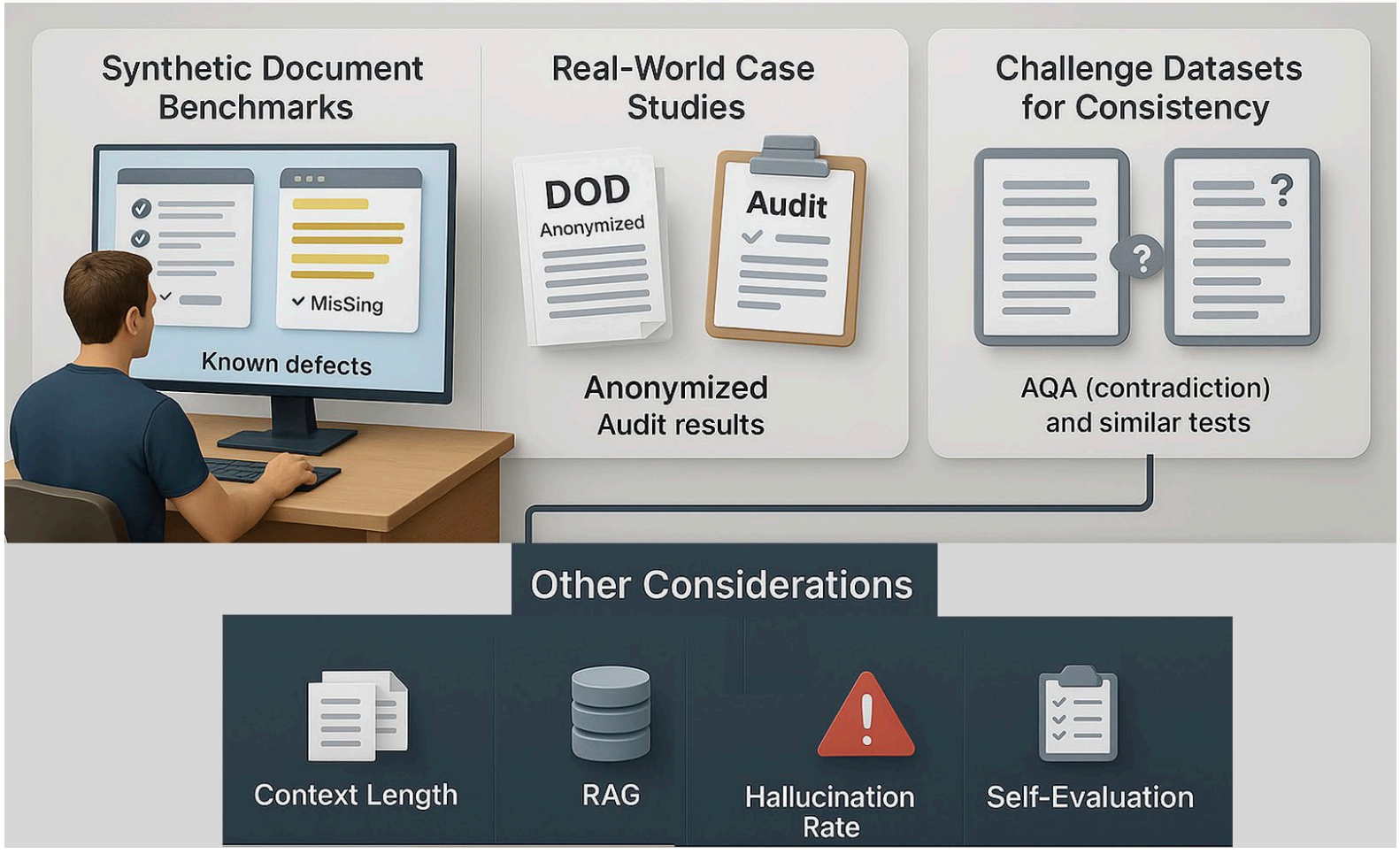




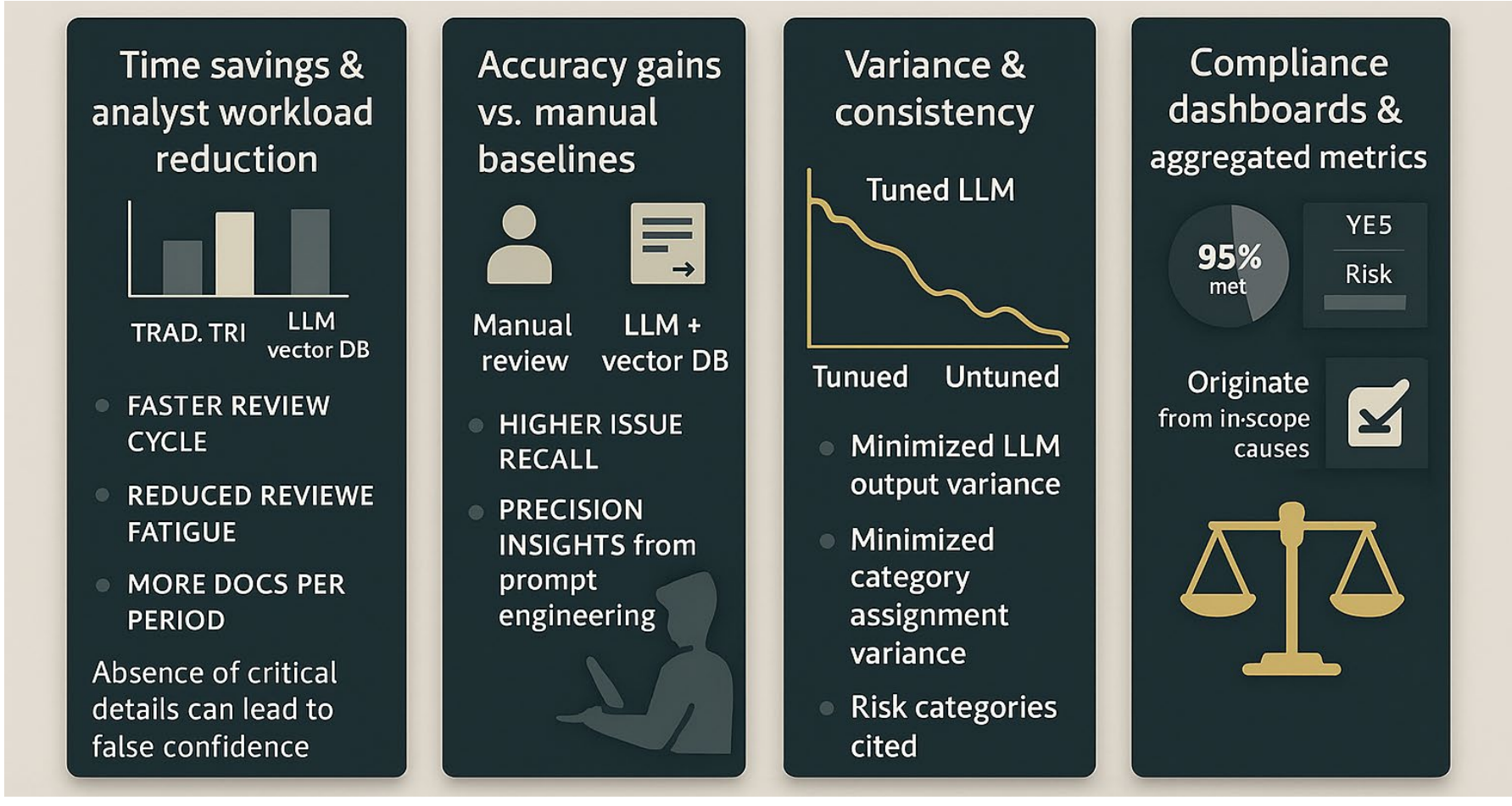
# The Role of Prompt Engineering in DIID Detection



# LLM Testing Frameworks for DIID Use Cases



# Measuring the Impact of AI-Augmented DIID Detection



# Future Research for AI-Augmented DIID Detection



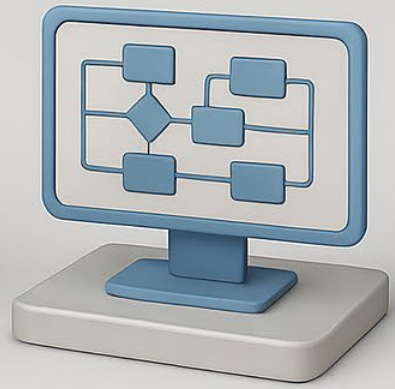
## Domain Benchmarking

- Benchmark LLMs across DoD domains
- Test general-purpose vs. fine-tuned vs. on-prem LLMs
- Metrics: adaptability, accuracy, Integration

## DIID Dataset Expansion

- Annotated document pairs (DIID issue label:)
- Expand domain-specific examples via LLM self-generation
- Semi-supervised augmentation with

## Multi-Modal DIID Detection



- Inputs: UMLs, ICDs, source code
- Use vision-language models or diagram parsers
- Compare diagram logic vs. narrative text for mismatc-

## Human-in-the-Loop Validation

- AI flags ambiguous/ conflicting clauses in policies
- Suggest edits or clarifications
- Feedback loop to UX & policy authors

# Thank You!


Picture (optional)