Introduction to Software Solutions Division

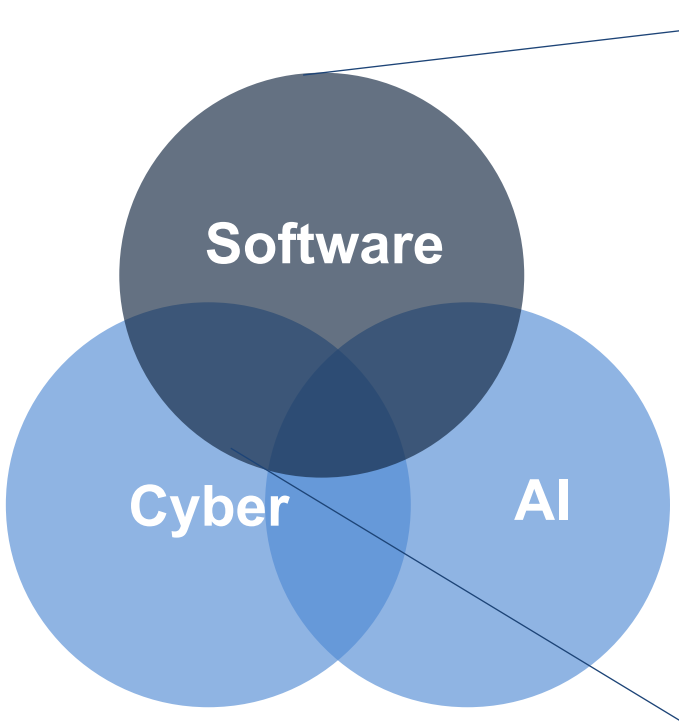
John Robert, Software Solutions Division, Deputy Director

December 2022

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.

DM22-0959

Software Engineering Strategy:
Rapidly Deploying Software Innovations with Confidence in DoD



SEI Strategic Framework

Exploit emerging technologies

- Engineering Intelligent Software Systems
- Enabling Mission Capability at Scale
- Assuring Cyber-Physical Systems
- Continuous Deployment of Capability
- Transforming Software Acquisition Policy and Practice

Curate & analyze data for decision-making

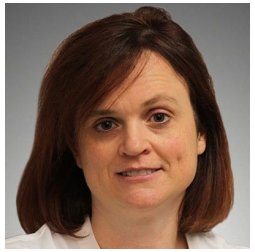
Software Solutions Leadership Team

Harold Ennulat
Associate Director of Client Integration and Program Development

John Robert
Deputy Director

Anita Carleton
Director

Mark Klein
Principal Technical Advisor



Erin Harper
Strategic Communications Manager

Technical Directors

Dr. Dionisio de Niz

Assuring Cyber-Physical Systems

Dr. Ipek Ozkaya

Engineering Intelligent Software Systems

Daniel Plakosh

Enabling Mission Capability at Scale

Eileen Wrubel

Transforming Software Acquisition Policy & Practice

Hasan Yasar

Continuous Deployment of Capability

Software Solutions Technical Directorates and Initiatives

Technical Directorate/ Director	Initiative Name and Lead	Initiative Description
Engineering Intelligent Software Systems Dr. Ipek Ozkaya	**Architecture Design, Analysis, & Automation**	Applying AI to automate architecture design and analysis activities; applying and accelerating adoption of architecture practices
	Tactical and AI-enabled Systems	Developing software engineering principles and practices for tactical and AI-enabled systems; advanced prototyping of the application of principles and practices
Enabling Mission Capability at Scale Daniel Plakosh	**Systems and Software Development & Analysis**	Scaling software development and deployment through AI/ML and automation for large, complex, real-time mission systems
	Resilient Critical Software Systems	Leveraging threat information and real-time software techniques to drive resilient solutions for mission critical embedded systems
	Advanced Deterrents	Contribute to the development of America's new 21st century deterrent platforms and weapons which will serve as the backbone of our nation's national security
Assuring Cyber-Physical Systems Dr. Dionisio De Niz	**Formal Verification of Cyber-Physical Systems**	Rapid and scalable automatic verification of cyber-physical systems built from verified and unverified components, ensuring outputs with the right value, at the right time, and with the right physical reaction (e.g., stop a crash)
	Model-Based Software Engineering	Virtual integration to discover flaws before implementation
Transforming Software Acquisition Policy and Practice Eileen Wrubel	**Software Acquisition Pathways**	Assembling modern software acquisition/development approaches to inform policy and practice
	Software Engineering Measurement & Analysis	Data analytics to drive software acquisition policy and priorities
Continuous Deployment of Capability Hasan Yasar	**Agile Transformation**	Modernizing software development and acquisition with Agile methods
	DevSecOps Innovations	Engineering for automated secure deployment and operations pipeline

Cyber-Physical Systems Assurance

Automated (Code and System) Testing and how that plays into overall SW development

- Model-based Test Generation
  - Test automatically generated from architectural models to validate key properties
- Shift Testing and Validation Left:
  - Start with early and incremental models
    - Analyzed models before implementation (or modifications to implementation) with automated methods
    - Automating implementation conformance through code generation / code verification
    - Connected to claims to validate authority to operate
  - Incremental model-analyze-build cycles

AI/ML Safety Verification

- Enforce AI/ML a safety envelope (with respect to safe values at a safe time)
  - Enable AI/ML to freely work within a safety envelope
- Verify that the enforced system preserve safety conditions
- Protect enforcement with respect to both failures and cyber-attacks

Tactical and AI-Enabled Systems (TAS)

We investigate and develop techniques to address challenges at the tactical edge:

- Limited computing resources at the edge for execution of AI/ML components
- Integrating AI/ML components in a pipeline that runs across multiple edge nodes
- Identifying the optimal tradeoff between accuracy, resource consumption, and inference speed
- Summarizing/filtering data for end users (e.g., operators) and deciding what data to store where
- Balancing microcontroller low memory/storage with desired capabilities

Embedded Software/Cyber Resilience

Challenge Areas

- Traditional cyber controls are not always applicable for embedded software with deterministic, mission critical processing needs
- RMF (NIST 800-37) provides an overarching framework for mitigation controls for cyber protection, however the available controls for embedded systems are limited
- SEI has developed focused and practical methods to understand risk areas and implement controls for mission critical software applications

SEI developed a three-pronged approach to enhanced software/cyber controls for mission critical, deterministic systems

1. Embedded software structural analysis to identify areas of high risk for exploitation, security, architecture limitations, testability, and maintainability. We call this process CREW
2. Attack surface and threat informed modeling to understand adversary activities
3. Develop methods to detect anomalies or intrusion within the software control system

Heterogeneous Computing

New computing technologies have significant differences

- Foundational computational approach
- Algorithmic solutions
- Processing to power ratios
- Error processing (fault detection and handling)

New software engineering approaches to integrate multiple computing devices

- Architecting integrated systems with predictable behaviors
- Interoperable tools for software debugging, profiling, and testing

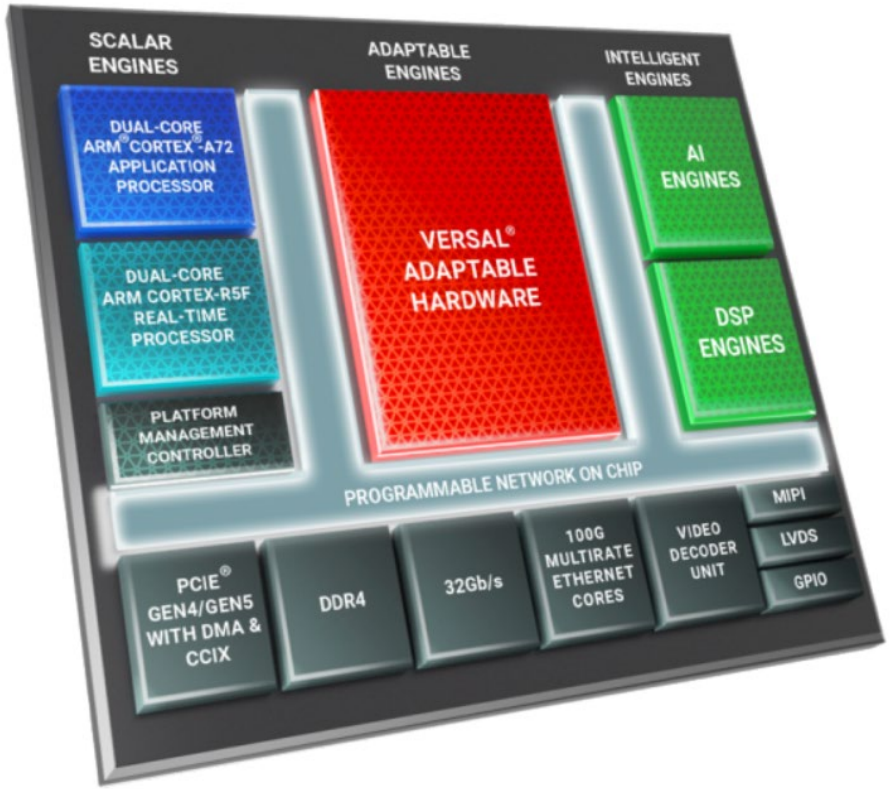


Image of Adaptive Compute Acceleration Platform (ACAP) from https://www.xilinx.com/products/silicon-devices/acap/versal-ai-core.html

National Agenda for Software Engineering R & D Study

The SEI led the community in creating this multi-year research and development vision and roadmap for engineering next-generation software-reliant systems.

Study released November 2021

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

Focus of National Agenda for Software Engineering

Software is vital to America's **global competitiveness, innovation**, and **national security**. The economy, the nation's infrastructure, education, and healthcare all depend on software.

Lead a community effort to:

1. **Identify future challenges in engineering software-reliant systems.**
2. **Develop a research roadmap that will drive advances in foundational software engineering principles across system types such as intelligent, autonomous, safety-critical, and data intensive systems.**
3. **Raise the visibility of software to the point where it receives the sustained recognition commensurate with its importance to national security and competitiveness.**
4. **Enable strategic partnerships and collaborations to drive innovation among industry, academia, and government.**

Foreword, National Agenda for Software Engineering

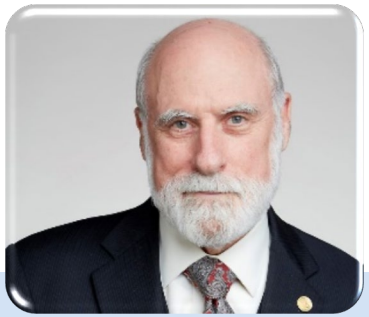
"Software is an essential, if not the central, part of every Department of Defense (DoD) system. Our hardware has become increasingly programmable, and software has become ubiquitous. Therefore, software engineering is a critical enabler for everything that we do in the DoD. To remain competitive, our weapon systems acquisition must migrate away from the linear development and test cycle and evolve into a rapid continuous update and continuous assurance environment. Consequently, this software engineering technology roadmap is a guide for our research and investment strategy that is vital for our national security. As we develop new systems, we must go beyond model-based software engineering to enable us to rapidly develop systems while reducing re-assurance and sustainment costs. In the future, we will need rapid composition of new capabilities that can operate in a highly contested and denied environment. Integrating heterogeneous systems seamlessly and rapidly will enable us to stay ahead of threats. We will need to exploit the promise of artificial intelligence to increase capability not only in our fielded systems but also in our development systems. This research roadmap should serve as the starting point for a sustained effort to improve software engineering. The DoD will continue to look to the Carnegie Mellon University Software Engineering Institute as a leader in improving the state of the art and practice in software engineering."

--The Honorable Heidi Shyu, Under Secretary of Defense for Research and Engineering

Guided by an Advisory Board of Visionaries and Senior Thought Leaders

Dr. Deb Frincke
Advisory Board Chair
Associate Laboratory Director, DOE Oak Ridge National Laboratory

Dr. Michael McQuade
Carnegie Mellon University Vice President for Research



Dr. Vint Cerf
Vice President and Chief Internet Evangelist for Google

Ms. Penny Compton
Vice President for Software Systems, Cyber, and Operations, Lockheed Martin Space

Mr. Tim Dare
[Previous] Deputy Director for Prototyping & Software, OUSD (R&E); Lead of NDAA Section 255

Mr. Jeff Dexter
Senior Director of Flight Software & Cybersecurity, SPACEX

Dr. Yolanda Gil
President, Association for the Advancement of Artificial Intelligence

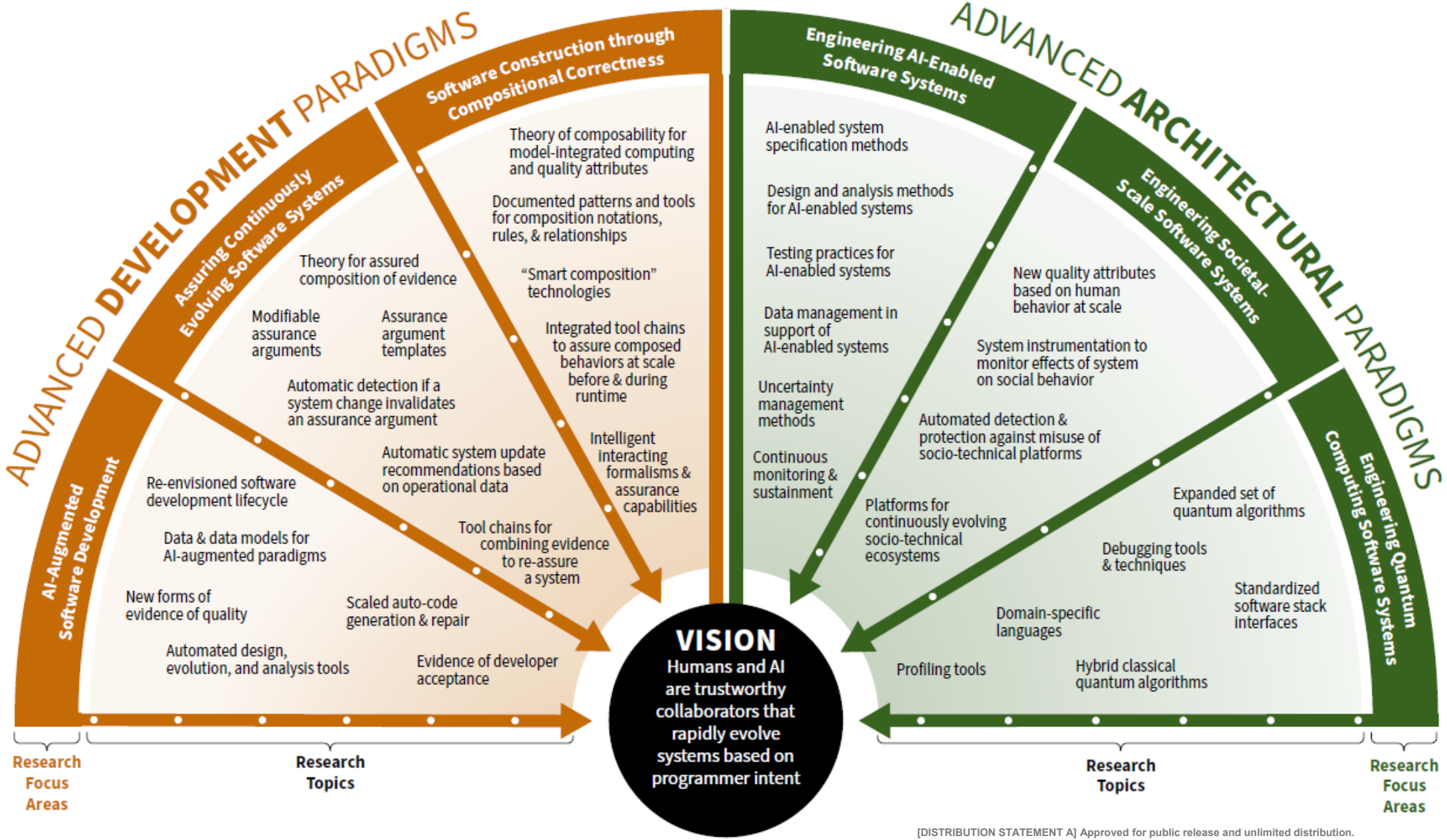
Ms. Sara Manning Dawson
Chief Technology Officer, Enterprise Security, Microsoft

Mr. Tim McBride
President, Zoic Studios (a leading visual technology firm supporting the DoD)

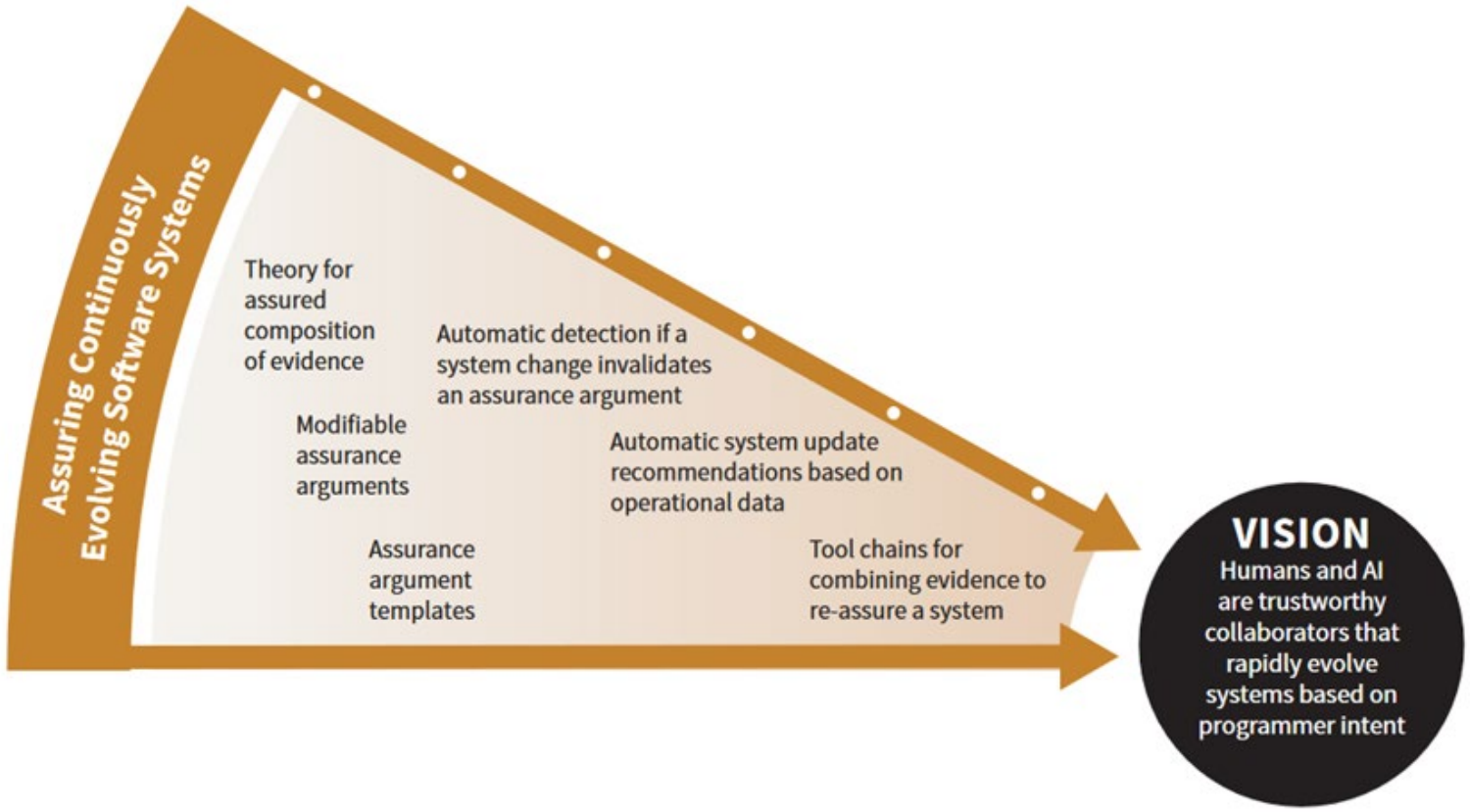
Ms. Nancy Pendleton
Vice President and Senior Chief Engineer, Boeing Defense

Dr. William Scherlis
Director, DARPA Information Innovation Office

Software Engineering Research Roadmap (10-15 Year Horizon)



Assuring Continuously Evolving Systems

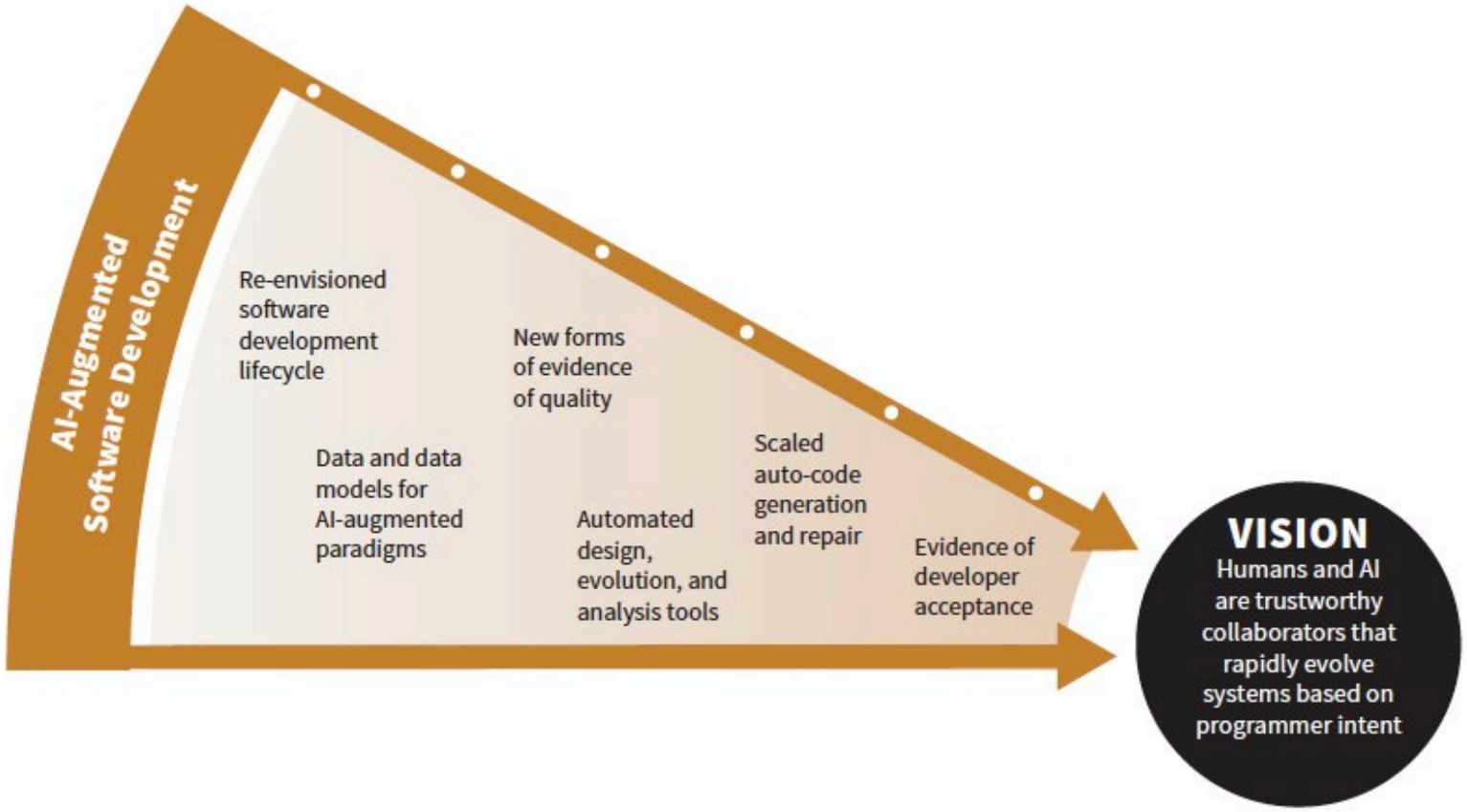


This research area focuses on developing a theory and practice of

- rapid and assured software evolution
- based on assurance arguments that "prove" that a system will behave as intended
- considering both desired functionality and quality attributes,
- as it evolves continuously to incorporate new capabilities and
- dynamically self-adapts at runtime in response to changing mission demands and environmental conditions.

Discussion

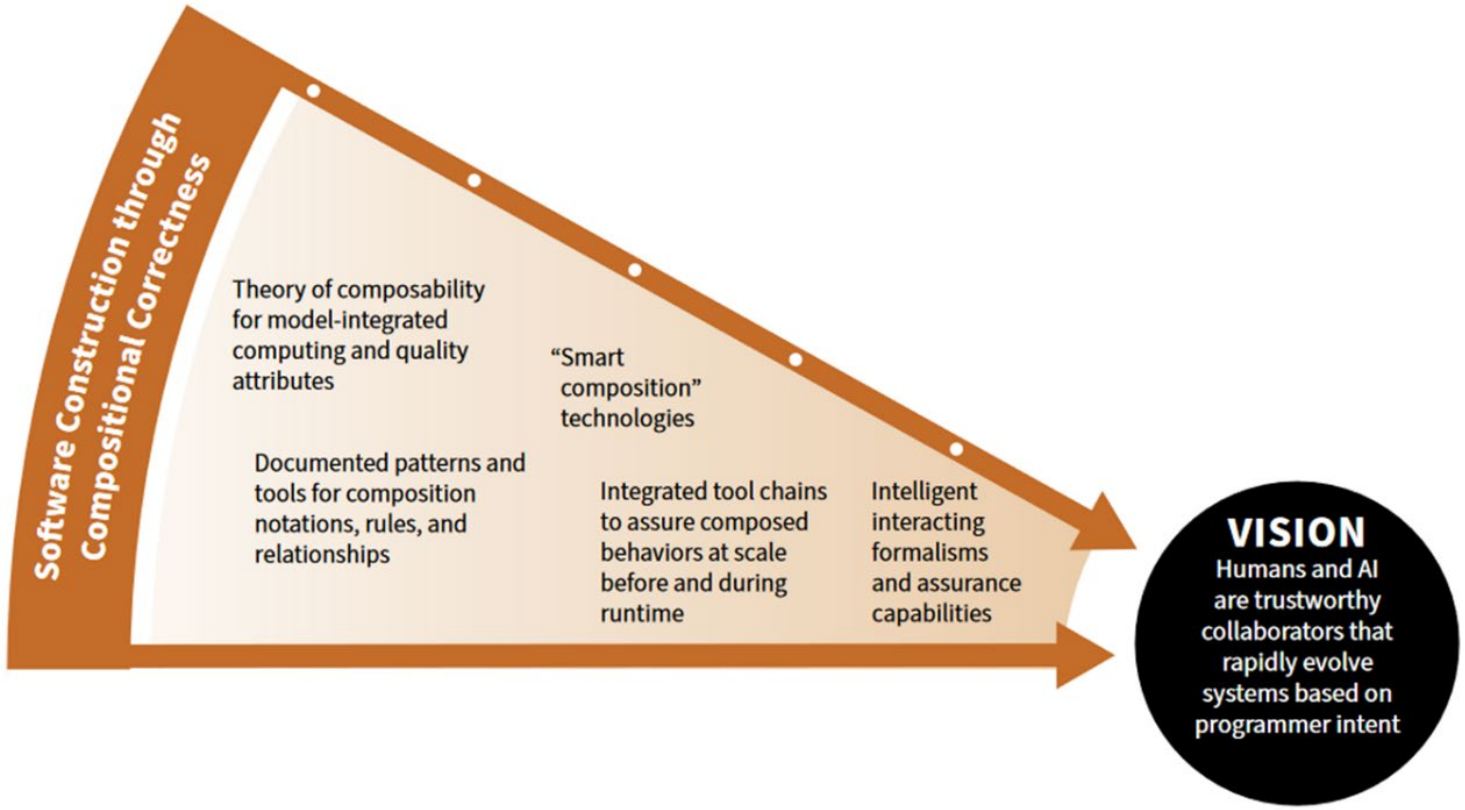
AI-Augmented Software Development



Are we providing effective tools to improve developers' tasks and cognitive overload towards developing higher quality software?

What will application of AI help solve that other approaches to date have not been able to help improved automated support for developers?

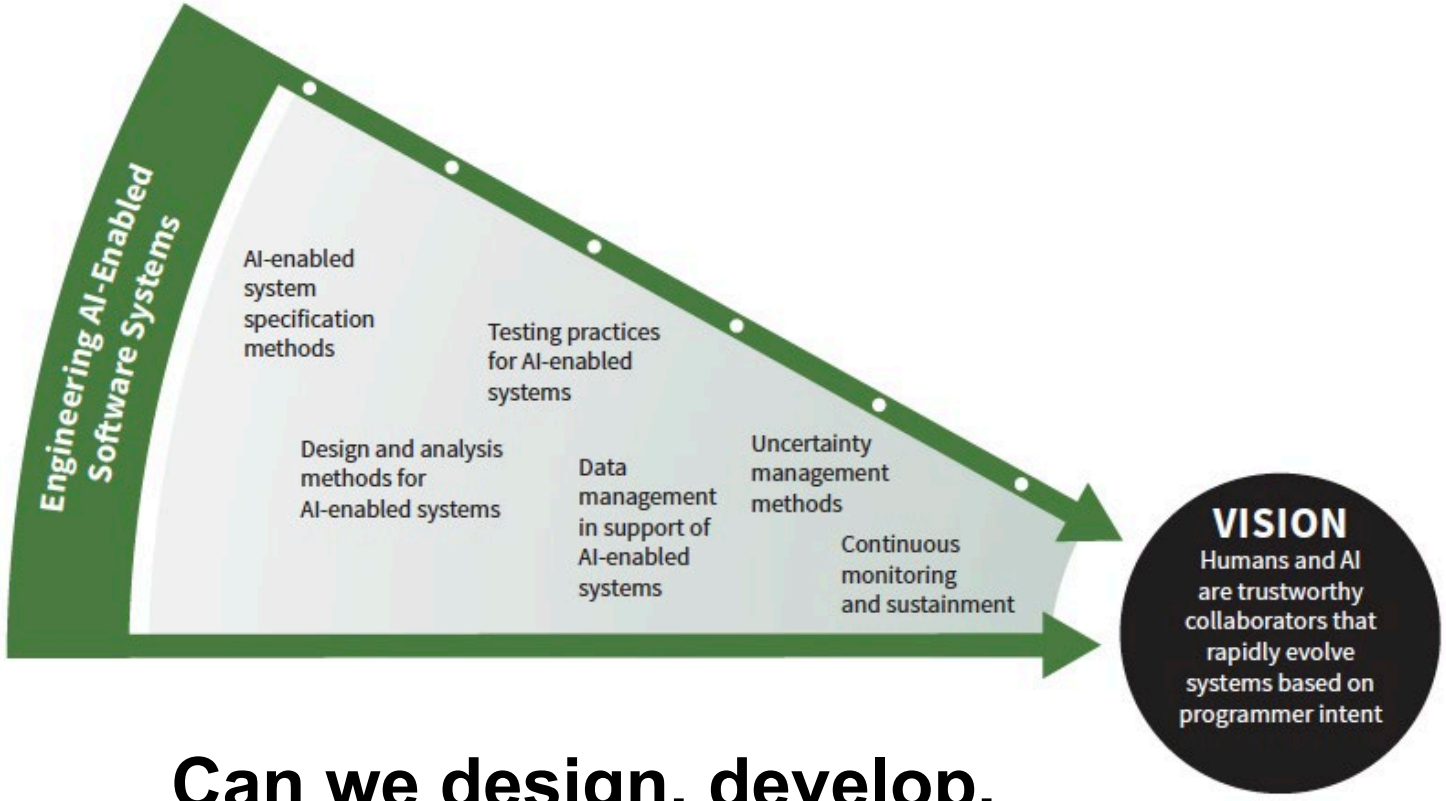
Software Construction Through Compositional Correctness



To address challenges associated with scale, complexity, and time-to-market, software-reliant systems are increasingly developed using component-based technologies.

To ensure component-based systems meet their business, technical, and financial requirements and constraints, research is needed on theories of composability—along with the associated methods, platforms, and automated tools—to enable the specification and enforcement of composition.

Engineering AI-Enabled Software Systems



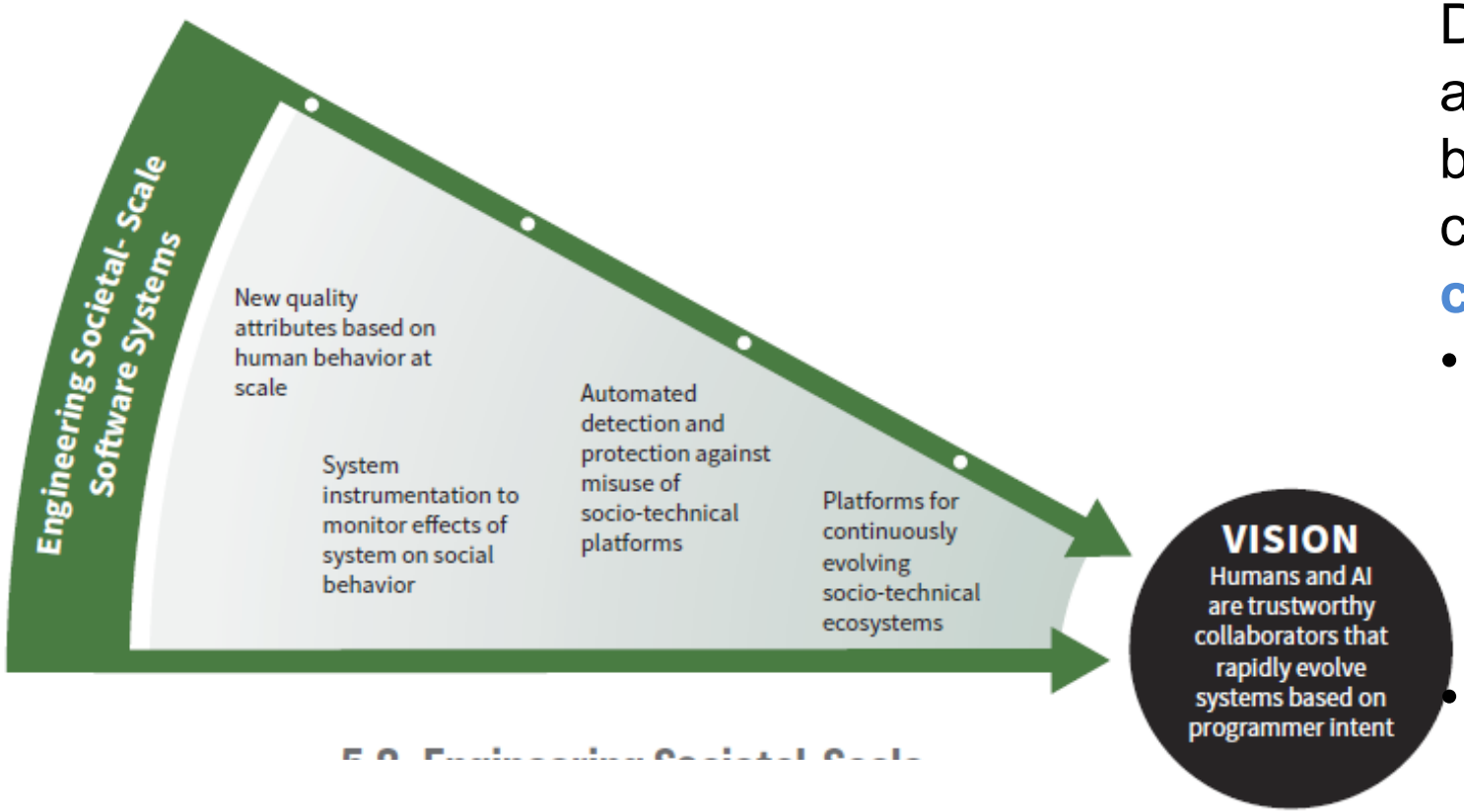
Can we design, develop, deploy and operated AI-enabled systems predictably?

Advances in ML algorithms and the increasing availability of computational power are already resulting in huge investments in systems that aspire to exploit AI.

- **Application of software engineering to AI problems**
- **Reinvigoration of data architecting**
- **Development of the new discipline of AI engineering will drive progress**

Studies increasingly are all emphasizing the disconnect between ML model development and operations of systems in the field (Lwakatare 2019, Serban 2020, Giray 2021)

Engineering Societal-Scale Software Systems

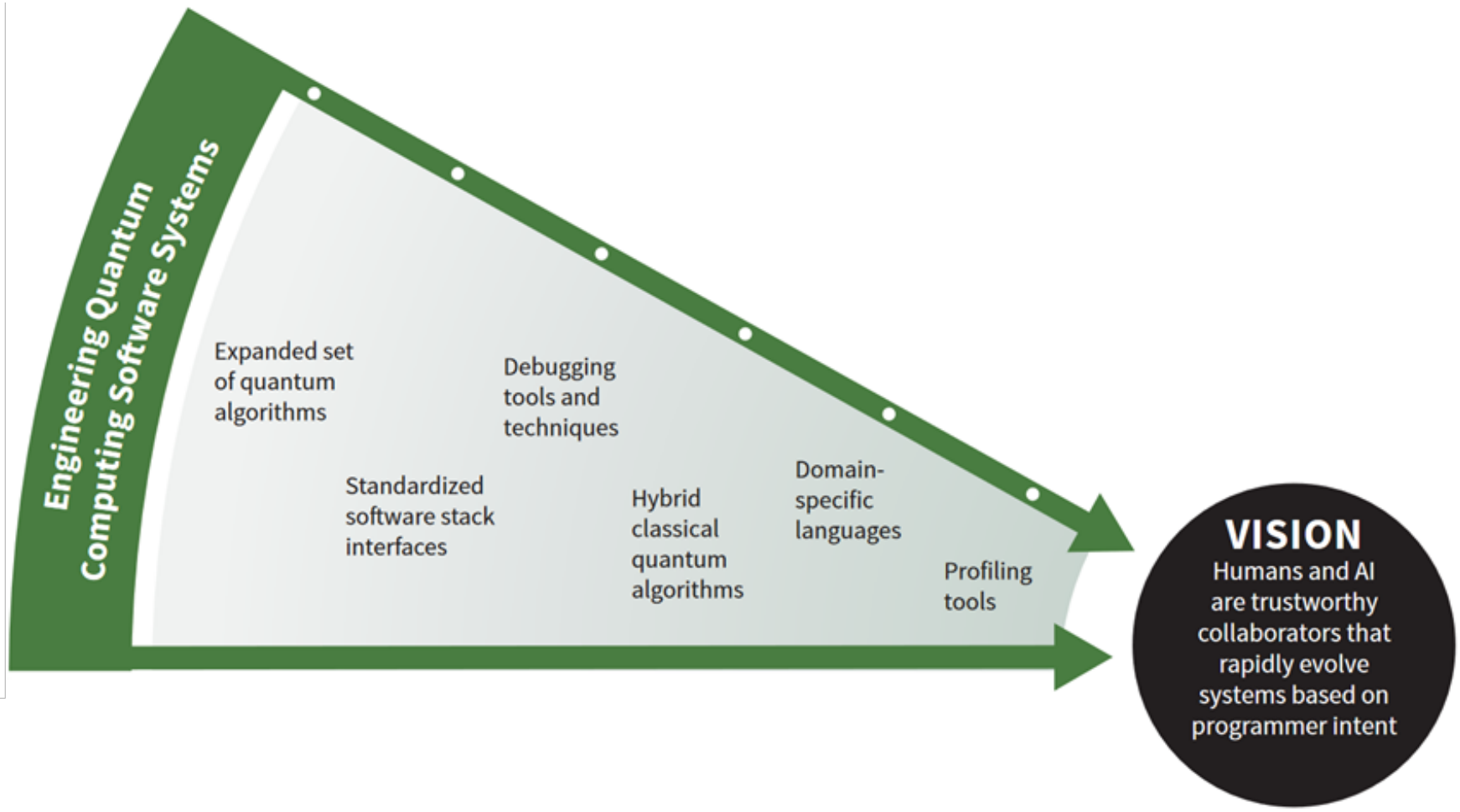


Develop new software engineering approaches that enable predictable behavior of software systems consisting of **people as system components**.

- Define foundational approaches that account for human behavior at scale with **self-reflection & correction** of continuously evolving socio-technical ecosystems.
- Enable the benefits of large societal systems, while **mitigating ethical risks** of privacy, bias, trust, concealed influence, or unrestrained social manipulation.

Ethics, bias, & misinformation—all are critical when **we depend on software**.

Engineering Heterogeneous Software Systems



Create approaches to integrate different types of computational devices into predictable systems.

Examples include computational devices based on

- **Multi-core**
- **Vector processing**
- **Neuromorphic solutions**
- Quantum mechanical effects