Research Agenda for Software Engineering
Architecting the Systems of the Future

July 8th, 2021

Dr. Douglas C. Schmidt

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded & 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 & development center.

Key Points in This Presentation

- Our ever-growing **dependence on software-reliant** systems makes it **imperative to continually innovate** in software engineering theories, tools, & practices
- Software is ubiquitous in today's systems—we must understand how to **develop & assure software continuously & insert new capabilities as quickly as possible.**
- **Future systems & fundamental shifts** in software engineering **require new research**.
  - e.g., rapidly deploying critical systems requires software engineering advances.
- Since new types of systems continue to push beyond what current software engineering theories, tools, & practices can support, **CMU's Software Engineering Institute (SEI) led a study to reevaluate & redefine how software is developed.**
- CMU SEI's study is intended to **catalyze the software engineering community** by creating a **multi-year research & development vision, strategy, & roadmap** for engineering next-generation software-reliant systems.

Objectives of Presentation

Discuss the Research Agenda Set Forth in the SEI’s Software Engineering Study

- Expected public release is August, 2021

Focus of Research Agenda for Software Engineering

Software is vital to **competitiveness, innovation**, & **security**. Global economies, infrastructure, education, entertainment, & healthcare all depend on software.

Improving software quality & productivity requires 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, & data intensive systems.
3. Raise the **visibility** of software to the point where it receives the sustained recognition commensurate with its importance to security & competitiveness.
4. Enable strategic partnerships & collaborations to **drive innovation among industry, academia, & governments**.

Abstract & Advisory Board

Software is vital to our country's global competitiveness, innovation, & national security. It also ensures our modern standard of living & enables continued advances in defense, infrastructure, healthcare, commerce, education, & entertainment. As the DoD's federally funded research & development center (FFRDC) focused on improving the practice of software engineering, the Carnegie Mellon University (CMU) Software Engineering Institute (SEI) is leading the community in creating this multi-year research & development vision & roadmap for engineering next-generation software-reliant systems.

Advisory Board

Deb Frincke, Chair
Associate Laboratory Director for National Security Sciences, Oak Ridge National Laboratory

Vint Cerf
Vice President & Chief Internet Evangelist, Google

Penny Compton
Vice President for Software Systems, Cyber, & Operations, Lockheed Martin Space

Tim Dare
Deputy Director for Prototyping & Software, Office of the Under Secretary of Defense for Research & Engineering; Lead of National Defense Authorization Act Section 255 (former)

Sara Manning Dawson
Chief Technology Officer, Enterprise Security, Microsoft

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

Yolanda Gil
Director of Knowledge Technologies, Information Sciences Institute, University of Southern California

Tim McBride
President, Zoic Labs

Michael McQuade
Vice President for Research, CMU

Nancy Pendleton
Vice President & Senior Chief Engineer for Mission Systems, Payloads & Sensors, Boeing Defense, Space & Security

William Scherlis
Director Information Innovation Office, DARPA

New System Types Require New R&D Advances

New types of systems

- highly adaptive mission systems
- systems that perform data fusion at a huge scale
- highly available business enterprises
- personal digital assistants—that really assist
- dynamically integrated, secure, & personalized holistic healthcare
- smart cities, buildings, roads, cars, & transportation systems

Trends

- scale motivating the need for safe & resilient software composition
- rapid & assured continuous software evolution
- addressing workforce gaps in software talent
- artificial intelligence (AI)-inspired automation
- evidence-based assurance that a system behaves as intended
- impact of cyber-social platforms on social behavior, creating societal scale impact
- primacy of data (data has become as important as code)

Summary of Our Study Approach

- Advisory Board
- Computing Landscape
- Emerging Technologies
- Literature Review
- Expert Interviews
- Workshops
- Future Scenarios

Cast a wide net with input from many communities

Research Agenda Study: Roadmap / Outcome

Codify findings in an actionable way

Diverse Ecosystem

Ecosystem acts on findings (hopefully!)

Emerging Vision of the Future of Software Engineering

The current notion of software development will be replaced by one where **the software pipeline consists of humans & AI as trustworthy collaborators that rapidly evolve systems based on user intent.**

Requires advances in **development paradigms** & **architectural paradigms**

Emerging Vision of the Future of Software Engineering

The current notion of software development will be replaced by one where **the software pipeline consists of humans & AI as trustworthy collaborators that rapidly evolve systems based on user intent.**

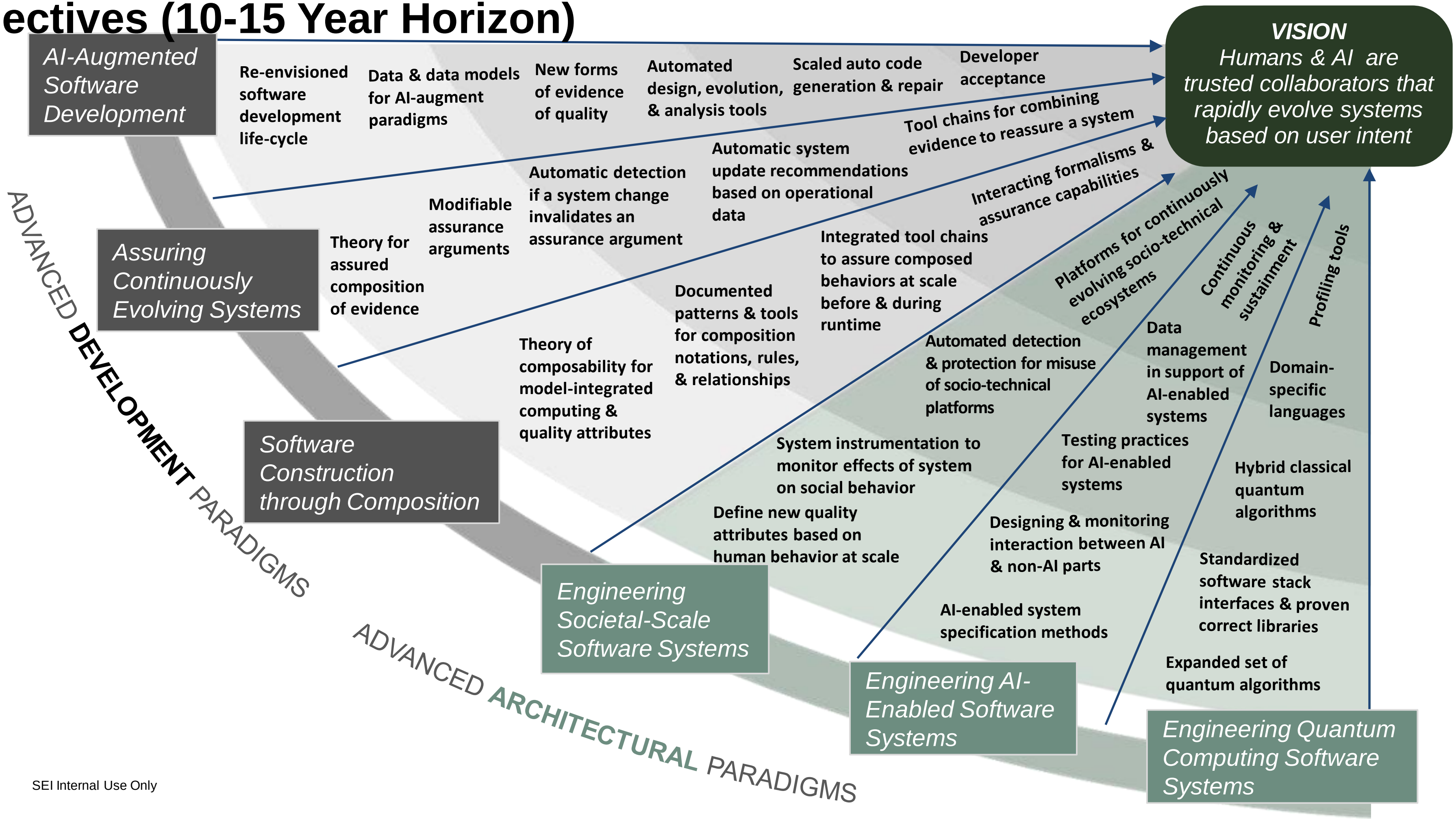
Advanced development paradigms lead to efficiency & trust at scale.

- Humans leverage trusted AI as a workforce multiplier for all aspects of software creation & sustainment.
- Formal assurance arguments are evolved to assure & efficiently re-assure continuously evolving software.
- Enhanced software composition mechanisms enable predictable construction of systems at increasingly large scale.

Advanced architectural paradigms enable the predictable use of new computational models.

- Theories & techniques drawn from behavioral sciences are used to design large-scale socio-technical systems, yielding more predictable outcomes.
- New analysis & design methods facilitate the development of quantum-enabled systems.
- AI & non-AI components interact in predictable ways to achieve enhanced mission, societal, & business goals.

Software Engineering Roadmap with Research Focus Areas & Research Objectives (10-15 Year Horizon)



Research Focus Areas: Development Paradigms

AI-Augmented Software Development

The focus of this research area is on what AI-augmented software development will look like at each stage of the development process & during continuous evolution, where it will be particularly useful in taking on routine tasks.

Assuring Continuously Evolving Systems

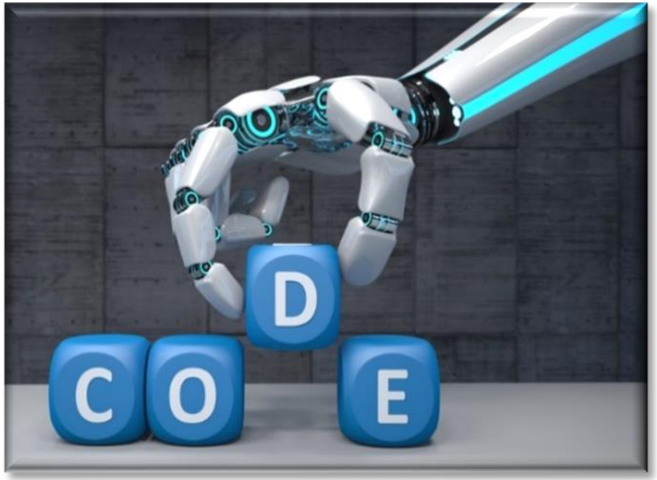
The goal of this research area is to develop a theory & practice of rapid & assured software evolution that enables efficient & bounded reassurance of continuously evolving systems.

Software Construction through Composition

This research area focuses on methods & tools that enable the specification & enforcement of composition rules for component-based technologies & platforms that allow both the creation of required behaviors & the assurance of these behaviors.

AI-Augmented Software Development - 1

Leverage advances in deep learning & search-based algorithms to support development of next-generation software design, evolution, & conformance tools that can enforce sound software engineering principles into system development.



- Augment each stage of software development with AI to orchestrate rapid development, continuous systems evolution, & rapid deployment of software-intensive systems.
- Orchestrate continuous evolution by distributing peer roles & responsibilities appropriately between human-led tasks & AI.
- Develop techniques to understand human expression of intent.

AI-Augmented Software Development - 2

Open research challenges for AI-Augmented Software Development:

- Designing new phases or activities, or re-design the existing activities, to re-think the software development life-cycle in an AI-Augmented paradigm.
- Incorporating the elicitation of user intent.
- Obtaining data to model each stage/workflow of an AI-Augmented paradigm.
- Identifying roles humans & AI can perform effectively so AI is a trustworthy peer.
- Automatically accumulating & carrying along evidence of quality; Verifying results are correct—use AI to generate meta-data needed to efficiently verify or validate code; generate proof with code.
- Determining how AI will orchestrate continuous systems evolution (see next area)

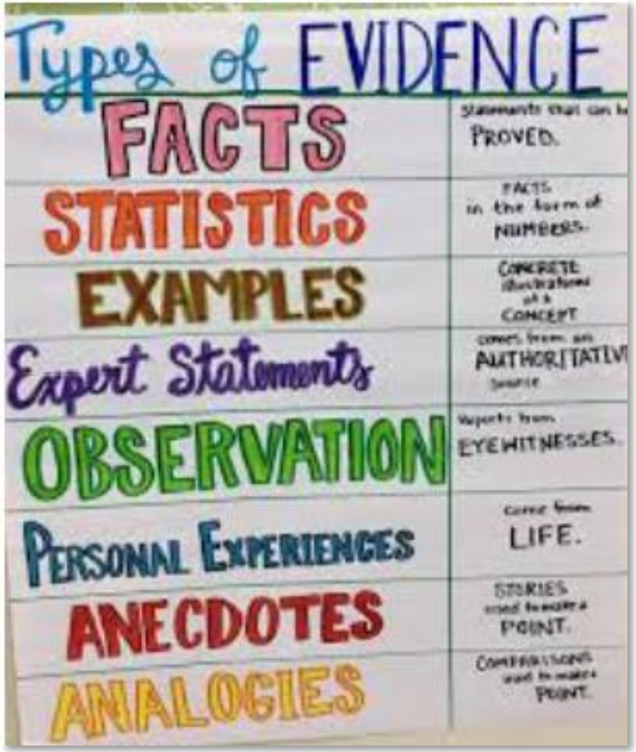
Assuring Continuously Evolving Systems - 1

This research area focuses on providing **evidence-based** assurance arguments that a system will behave as intended as it **evolves** continuously to incorporate new capabilities & dynamically self-adapts its operating configuration(s) at runtime in response to changing mission demands & environmental conditions.

- Must consider both desired **functionality** & **quality attributes.**

This research area requires advances in

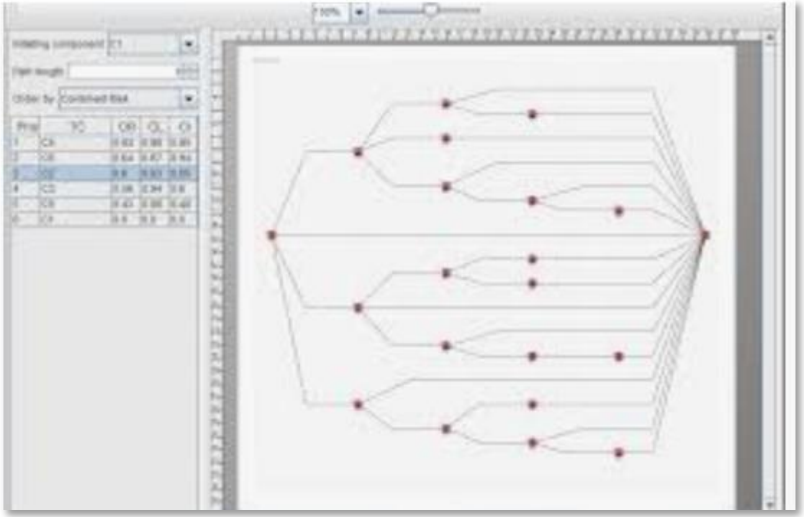
- "Theory of evidence"
- "Theory of evolution"



Assuring Continuously Evolving Systems - 2

"Theory of evidence" for

- combining different types of evidence & …
- automatically creating (semi-)formal arguments or assurance cases



"Theory of evolution" for

- understanding how potential system changes propagate & possibly lead to new failure modes or vulnerabilities
- automatically generating & analyzing runtime data & generate system improvement suggestions
- combining human expression of intent & with operation data to generate & assure change requests
- using AI to automatically generate evolution trajectories

Software Construction Through Compositional Correctness -1

This research area focuses on developing software -reliant systems built from modular components, where behaviors & quality attributes of **component compositions should be more robust & resilient than component parts in isolation**.

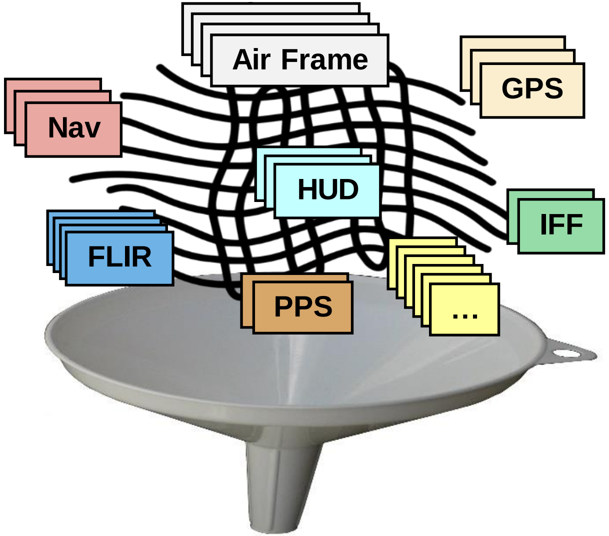
Achieving success in this area requires a new theory of composition & associated patterns & integrated tool chains that can assure the following properties.

- The effort required to (re)assure & (re)certify an entire system as it evolves should be minimized
- The complexity & cost of assurance should grow no more than linearly with the size of the system & the scope of the changes.

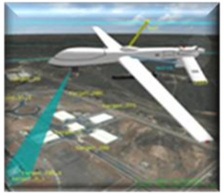
Software Construction Through Compositional Correctness - 2

Advanced capabilities needed to support compositional correctness at scale:

- A theory of composability that supports cross-cutting concerns & quality attributes.
- Documented patterns & tools that enable the specification & enforcement of composition rules.
- Integrated tool chains that create required behaviors & assure these behaviors at scale.
- Formalized interactions & non-siloed assurance capabilities supported by integrated tool chains.



ISR Processing



Air Traffic Mgmt

Aerospace

Engineering Socio-Technical Systems - 1

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

- Societal-scale systems (such as social media platforms) define foundational approaches that **account for human behavior at scale** with self-reflection & correction of continuously evolving socio-technical ecosystems.

- This research area focuses on building & evolving societal-scale software systems that **enable the benefits** of large societal systems, **while mitigating ethical risks** of privacy, bias, trust, concealed influence, or unrestrained social manipulation.

Engineering Socio-Technical Systems - 2

New approaches needed

- Combining psychology, sociology, economics, & game theory techniques with software engineering to build-in new dimensions of system behaviors & ethics.
- Experimentation environment built into systems to explore & test new approaches at scale.

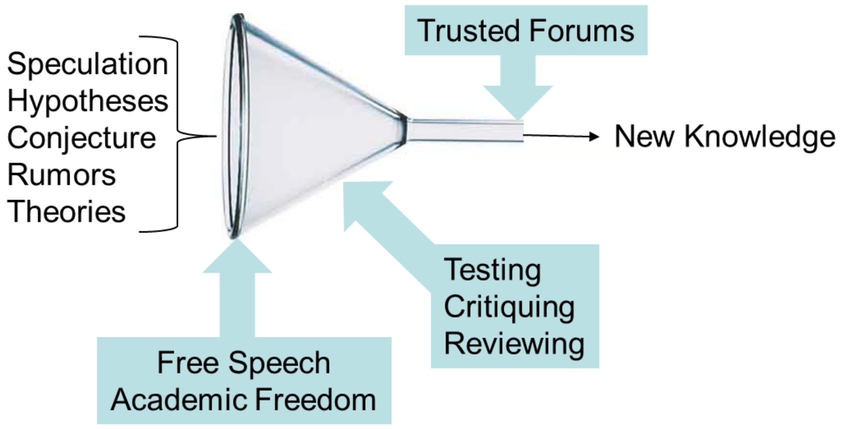


Image courtesy of Dr. Jim Herbsleb, Director CMU ISR

Key Concepts

- Define new quality attributes, architectures, & analysis tools that account for people.
- Develop a theory of socio-technical epistemology, i.e., what's a "justified true belief"?
- Balance tradeoffs between quality attributes, including conformance to policies.
- Automatically detect & protect against misuse of socio-technical platforms.

Summary of Study Findings

1. Software engineering **profoundly impacts all aspects of society** as we increasingly rely on it to provide complex & critical functionality.

2. **Key issues** impacting future directions in software engineering include **smart automation, reassuring evolving systems, understanding composed systems**, & **new system types**.

3. Research focused on **integrating heterogenous systems at ever-larger scales** is needed to support the emergence of new system types.

4. Beyond scale, tomorrow's software challenges will include the need to address **social software considerations** such as transparency, freedom from bias, & privacy.

5. **Academia, government, industry, & research labs** will have critically important and **interacting roles** for the future of software engineering.

6. Significant **research opportunities lie at the intersection** of software engineering and other fields.

7. Software engineering is a distributed activity, taking place worldwide. Technologies are needed to support **seamless integration across computing environments & teams.**

8. **New approaches** are needed to meet growing needs in the **workforce.**

Discussion