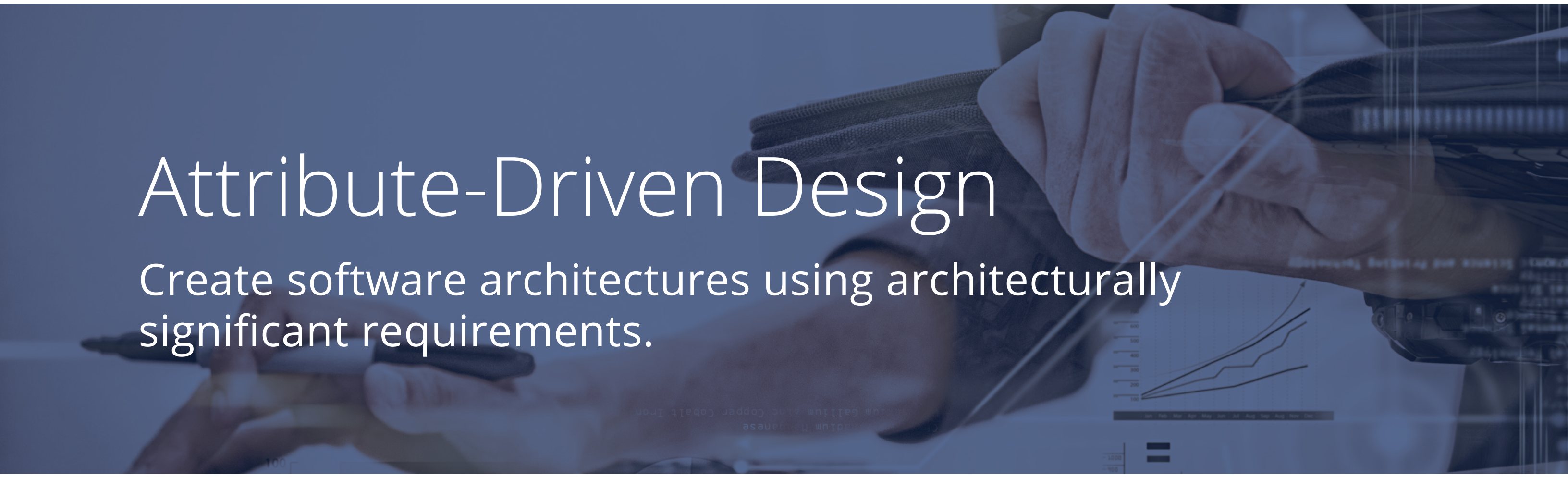
Carnegie Mellon University
Software Engineering Institute



Attribute-Driven Design

Create software architectures using architecturally significant requirements.

HOW DO YOU DESIGN A SYSTEM ARCHITECTURE THAT BEST MEETS YOUR USERS' NEEDS? How do you determine which architectural strategies to use for your new system?

Attribute-Driven Design (ADD) is a systematic, step-by-step method that helps you design an effective architecture for software-intensive systems. With ADD, your design process is based on the system's architecturally significant requirements (ASRs), which include functional requirements, quality attribute requirements, and constraints.

ADD can be used for application domains ranging from information systems to embedded systems.

The results of using ADD is a set of sketches of architectural views that a system designer can work from to develop the detailed architecture.

Challenges

ADD helps your organization tackle the following challenges:

- designing a system architecture that meets users' functional requirements
- meeting quality attribute requirements
- understanding the constraints that affect the design
- determining which architectural strategies are appropriate
- understanding the impact of quality attribute tradeoffs while designing a software architecture

Using ADD

ADD follows an iterative decomposition process where, at each stage in the decomposition, you choose tactics and architectural patterns to satisfy the ASRs. There are five steps in this process:

1. Select a component of the system that has not yet been designed. In the first iteration, this will be the system itself. With each successive iteration, you examine all components of the system you architecting.
2. Identify and prioritize the relevant ASRs that are most important for the selected component. This means that you select the functional requirements, quality requirements, and constraints that could affect the system architecture of the component you selected for this iteration of the process.
3. Build an architectural sketch for the component that meets the ASRs. This step is the heart of ADD since the architectural sketches. The sketches you create reflect the ASRs you selected. You use architectural patterns or tactics to guide your development, and you often make tradeoffs between tactics and patterns, and the ASRs they are satisfying.
4. Inventory the remaining ASRs and select a subset for the component to be architected in the next iteration.

Repeat steps 1-4 until all ASRs are resolved and you have an architectural sketch for each component.

Benefits

ADD enables you to understand quality attribute tradeoffs early in the design process and guides you in designing an architecture that satisfies the ASRs (the functional requirements, quality attribute requirements, and constraints). The output of the ADD is a set of architectural sketches that help you form the final architecture.

Get Started

SEI researchers can help you use ADD or answer technical questions about it. Contact us by calling 888.201.4479 or sending email to info@sei.cmu.edu.

Software Architecture Training

The SEI offers software architecture courses and certificate and certification programs that are based on extensive SEI and community experience in architecting software-intensive systems.

More than 20,000 people from more than 1,800 organizations have attended courses in the SEI Software Architecture Curriculum, and more than 2,500 people have earned software-architecture-related certificates.

Visit **www.sei.cmu.edu/education-outreach/courses/** to see the complete set of architecture-related offerings and register for upcoming courses.

SEI Expertise in Software Architecture

For more than two decades, the SEI has been instrumental in creating and developing the field of software engineering known as *software architecture*.

A system's software architecture is the conceptual glue that holds every phase of a project together for all of its stakeholders. It is the depiction of a system that aids in understanding how the system will behave.

Software architecture serves as the blueprint for both the system and the project developing it, defining the work assignments that must be completed. The architecture is also the primary carrier of system qualities such as performance, modifiability, and security, none of which can be achieved without a unifying architectural vision.

About the SEI

The Software Engineering Institute is a research and development center that works with defense and government organizations, industry, and academia to advance the state of the 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

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

W: sei.cmu.edu
T: 412.268.5800 | 888.201.4479
E: info@sei.cmu.edu