Integrated Safety and Security Engineering for Mission Critical Systems

Progress and planning in the first year of a three-year project

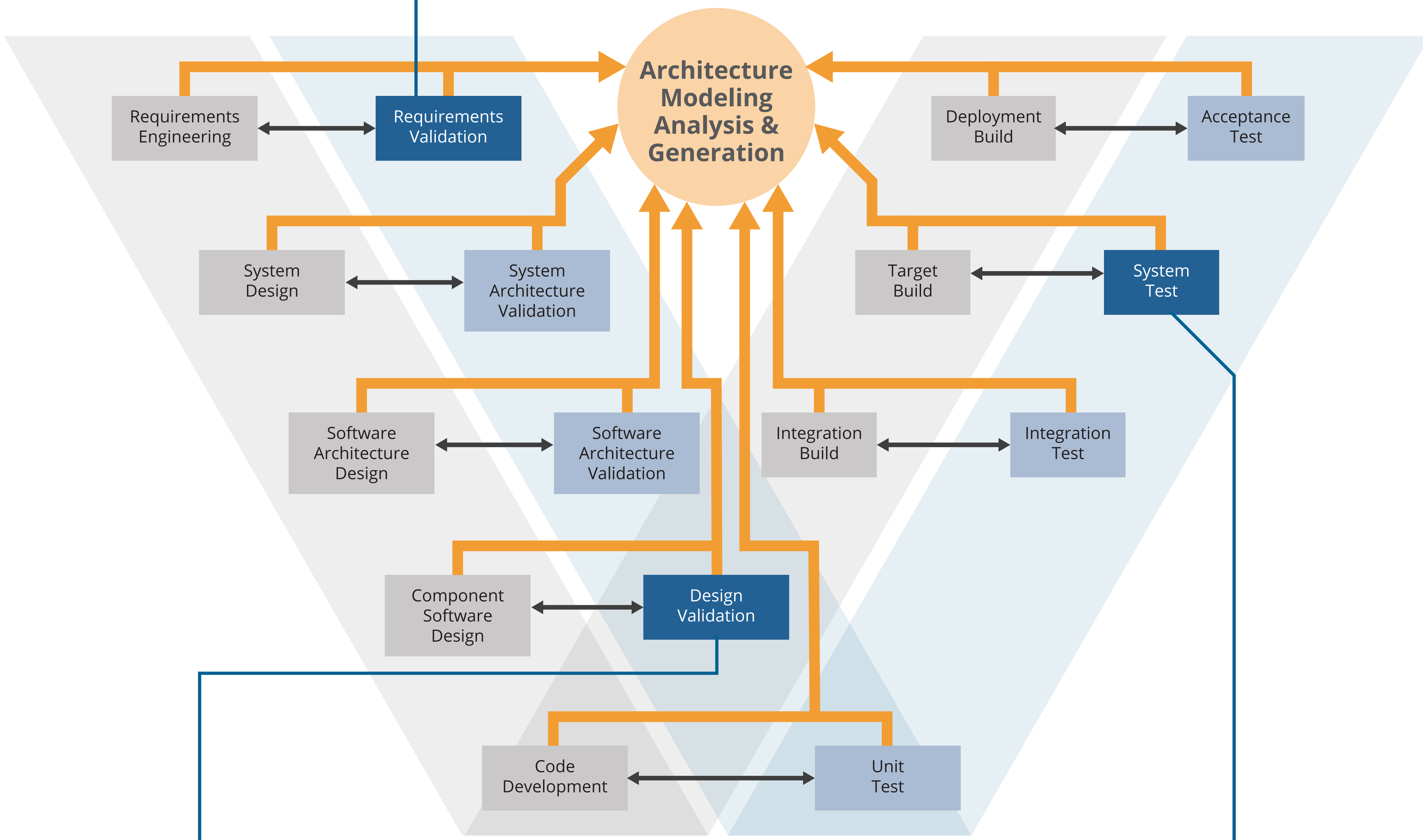
Safety-critical systems, such as airplanes and medical devices, are increasingly connected to the outside world. This adds new capabilities, but also exposes new security risks. We're looking at integrating security engineering techniques with safety processes using a system's architecture.

This work builds on years of successful research with AADL. Previously, the Architecture-Led Incremental System Assurance (ALISA) project established a toolkit and process for reasoning about safety throughout a system's development. Using this technology, we're creating guidance, examples, theory, and new tooling to guide developers of safety-critical systems to also reason accurately about security concerns.

Our development environment has tooling based on state-of-the-art hazard/threat-analysis theory. Previous work, both at the SEI and from the larger research community, has indicated that an effects-focused approach can offer a number of benefits for designing critical systems. Working with our collaborators, we're using this effects-focus to guide updates to our development environment, which is already being used in industry, commerce, and by a number of DoD contractors.

> The end result will be a tool-based, architecture-centric set of guidelines and automated analyses that brings security and safety together early in the system development lifecycle—avoiding costly and time-consuming rework.

We are identifying gaps in current architecture-centric security practices, such as poor documentation of a system's environmental assumptions. We are developing guidance, examples, and tooling to close those gaps. Where those practices conflict with safety guidance, we're documenting the tradeoffs so developers and stakeholders can be more informed.



Build the System

Build the Assurance Case

Along with our collaborators, we are developing a fault injection framework that will let us test a component's error behavior specification. This greatly simplifies testing components in exceptional conditions—currently a very challenging task.

Testing late in the development lifecycle is expensive, and fixes required at this point are similarly costly. This project, like its predecessor ALISA, shifts issues "to the left" so they can be addressed more quickly, cheaply, and—most important— effectively.

Sam Procter | sprocter@sei.cmu.edu | Peter Feiler | phf@sei.cmu.edu
External Collaborator: Kansas State University

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.

DM18-1139