

Integrated Safety and Security Engineering for Mission Critical Systems

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

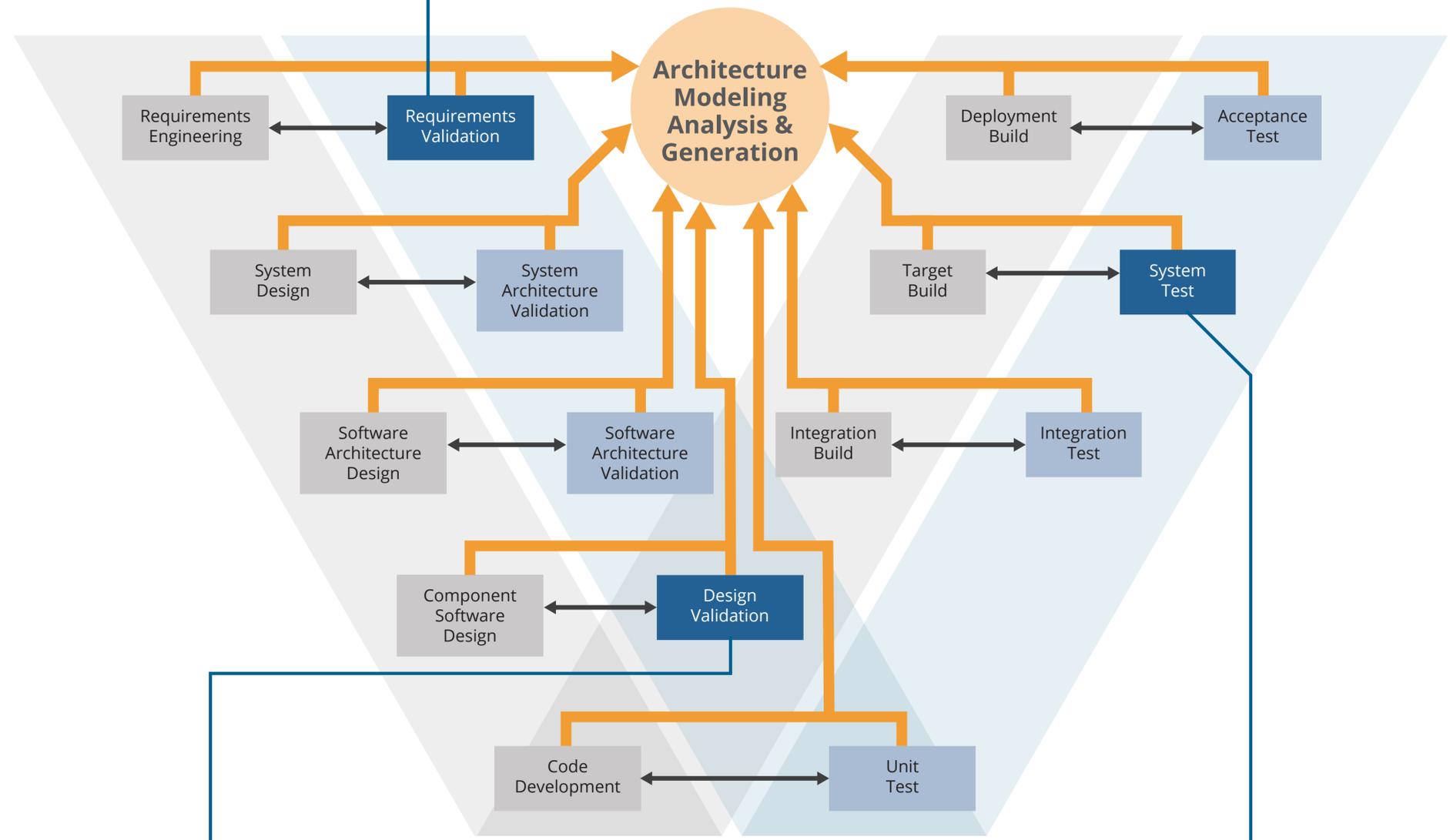
Safety-critical systems, such as airplanes and medical devices, are increasingly connected to the outside world. This adds new capabilities, but also exposes new security risks. We're looking at integrating security engineering techniques with safety processes using a system's architecture.

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.



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.

Copyright 2018 Carnegie Mellon University. All Rights Reserved.

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.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT. [DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM18-1139