

SYSTEM-OF-SYSTEMS SOFTWARE ARCHITECTURE EVALUATION

Software-intensive systems often suffer severe integration and operational/ behavioral problems due to a lack of consistency between the system and software architectures in addressing system quality attributes. These problems often result in rearchitecting and redesigning efforts and operational failures, which significantly affect system cost, schedule, and mission effectiveness. These problems are further exacerbated in a system of systems (SoS) context.

In conjunction with the Mission Thread Workshop (MTW), the SoS Architecture Evaluation Method provides an initial identification of SoS architectural risks and quality attribute inconsistencies across the constituent systems. An SoS architecture evaluation

- uses outputs of the MTWs, including augmented mission threads and SoS architecture challenges
- incorporates the expertise of a trained evaluation team and SoS stakeholders, including the SoS and system architects
- probes architecture at the areas where the systems interact to identify risks
- organizes the individual risks into risk themes that can be comprehended (and mitigated later) by program management
- assesses the sufficiency of architecture documentation
- identifies potentially problematic systems for focused follow-on evaluations using the specific augmented mission threads

The SoS Architecture Evaluation Method is ready to be piloted. If you are interested, contact us.

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479

Web: www.sei.cmu.edu | www.cert.org

Email: info@sei.cmu.edu

Copyright 2017 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.

DM17-0235