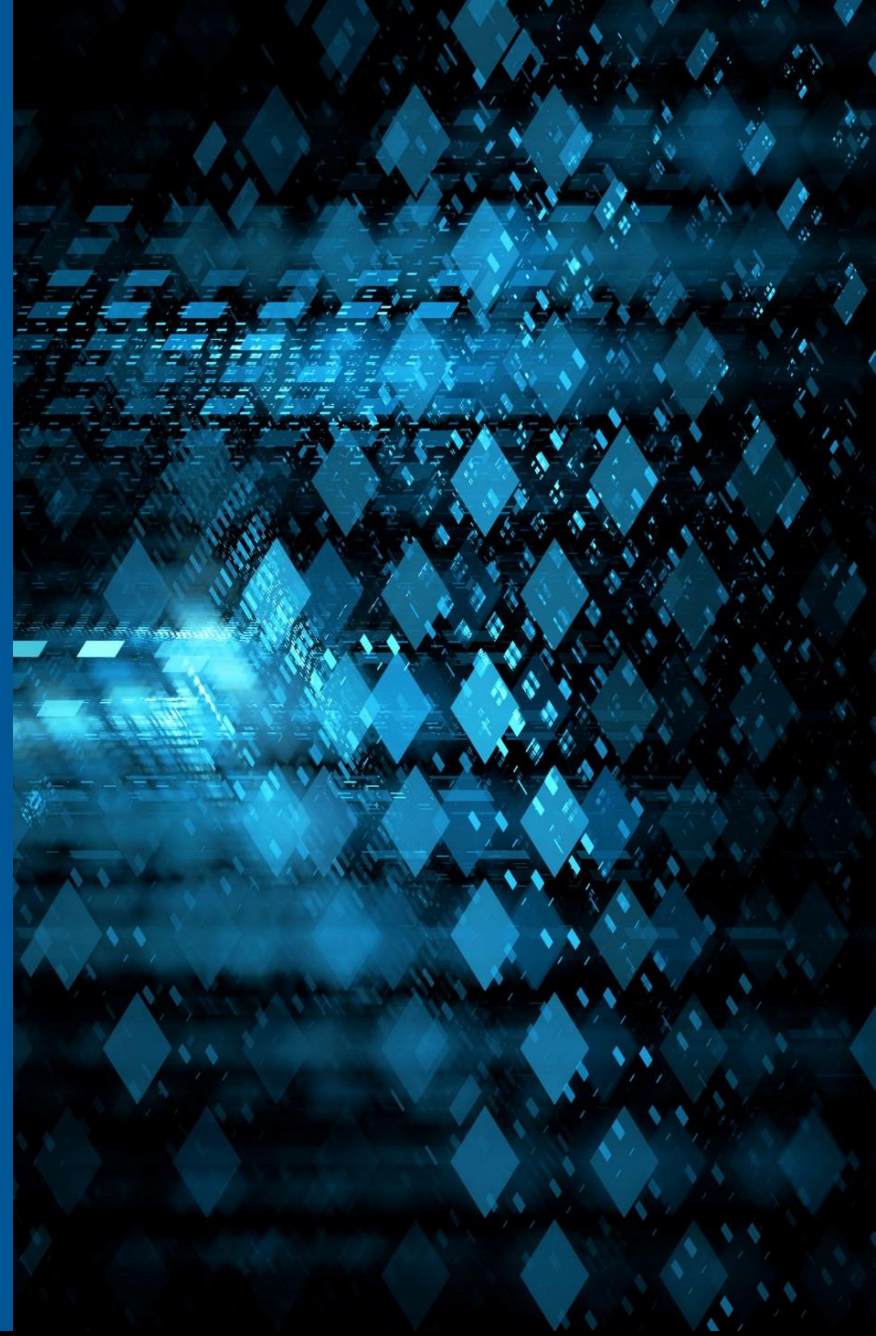
Driving Efficiencies into the Software Life Cycle for Army Systems

Stephen Blanchette Jr.

Presented to the CECOM Software Solarium
7 September 2016

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



© 2016 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.
REV-03.18.2016.0

Copyright 2016 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

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.

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 use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM-0003950

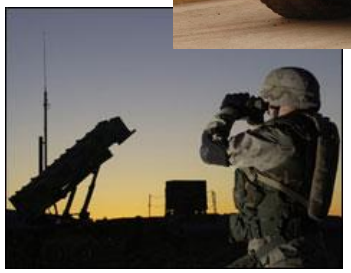
© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Where behavior counts most…

…much is required of software

© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

The Army's software challenges are many



Huge system/software engineering endeavors in weapon systems, command and control systems, enterprise systems, simulators, even ammunition

- Several million SLOC programs; hybrid systems combining legacy re-use, COTS, new development
- Multi-contractor teams using different processes; dispersed engineering, development & operational locations
- New technologies creating opportunities/challenges; products change/evolve, corporations mutate
- Business/operational needs change - often faster than full system capability can be implemented
- Skillset shortfalls; cost and schedule constraints
- Increased demands for integration, info/cyber-security, interoperability, system of system capabilities
- Software increasingly connects other systems (manned/unmanned teaming, swarming, etc.)
- Enterprise perspectives/requirements
- Growing (but late) concerns about sustainment

© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

We tend to make the same mistakes…

Requirements

- Undocumented assumptions
- Creep, instability
- Functionality – System Qualities = Incomplete Specs

Architecture – for software???

Assume testing yields quality

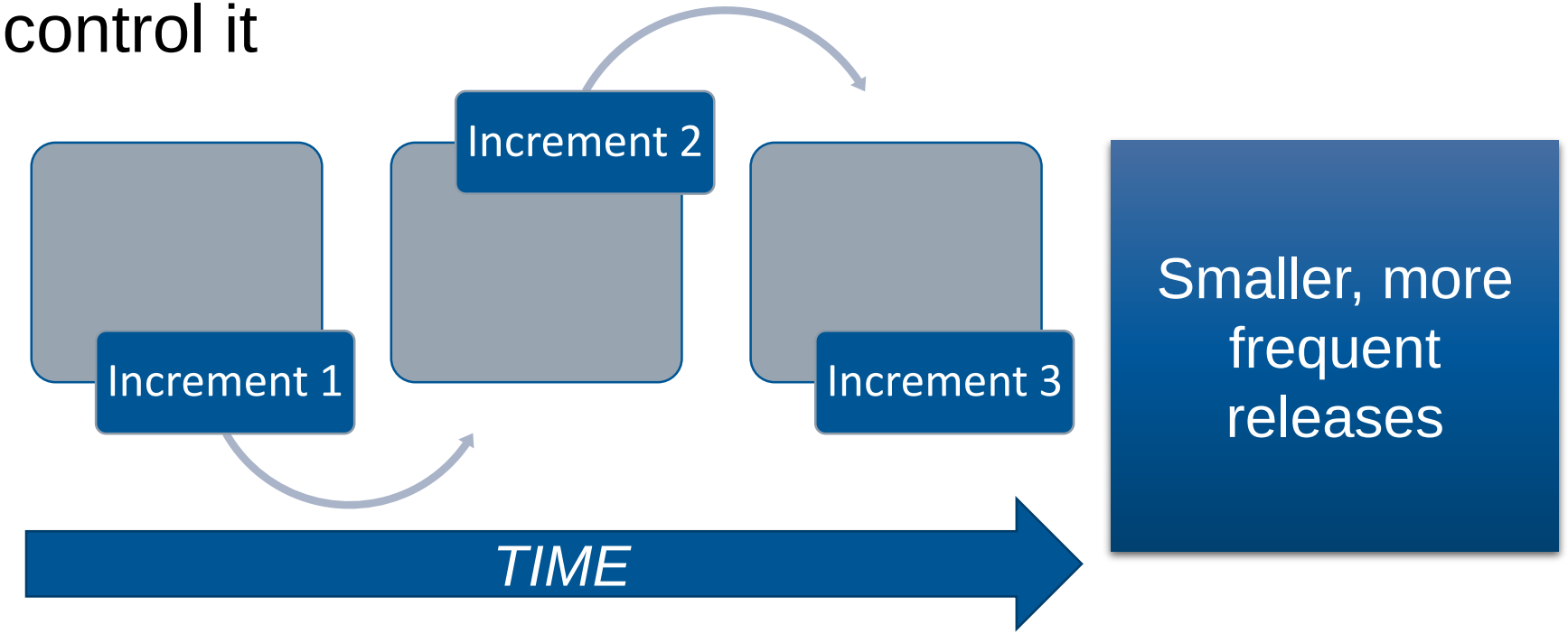
Ignoring and/or marginalizing the role of software from program outset

We can do better!

© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

We can use incremental methods to embrace change…

…but also control it

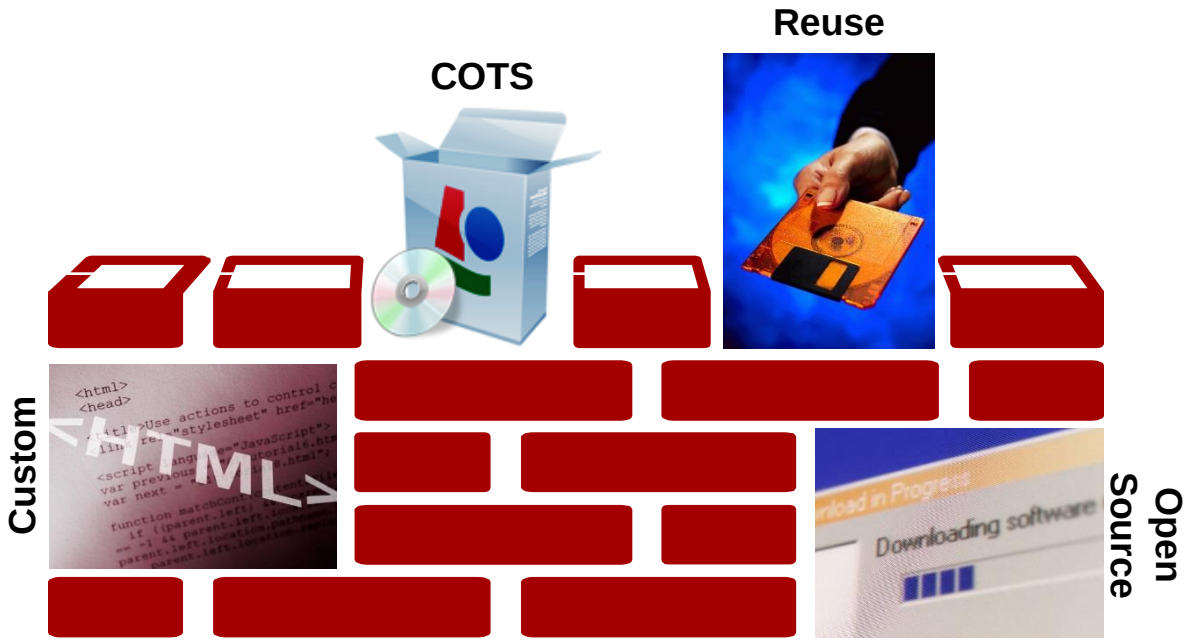


Examples include Spiral development, Incremental Commitment Model, Agile methods such as Scrum and XP, etc.

© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

We can employ software architecture to provide a foundation 1 of 2

To help plan out how the pieces go together…

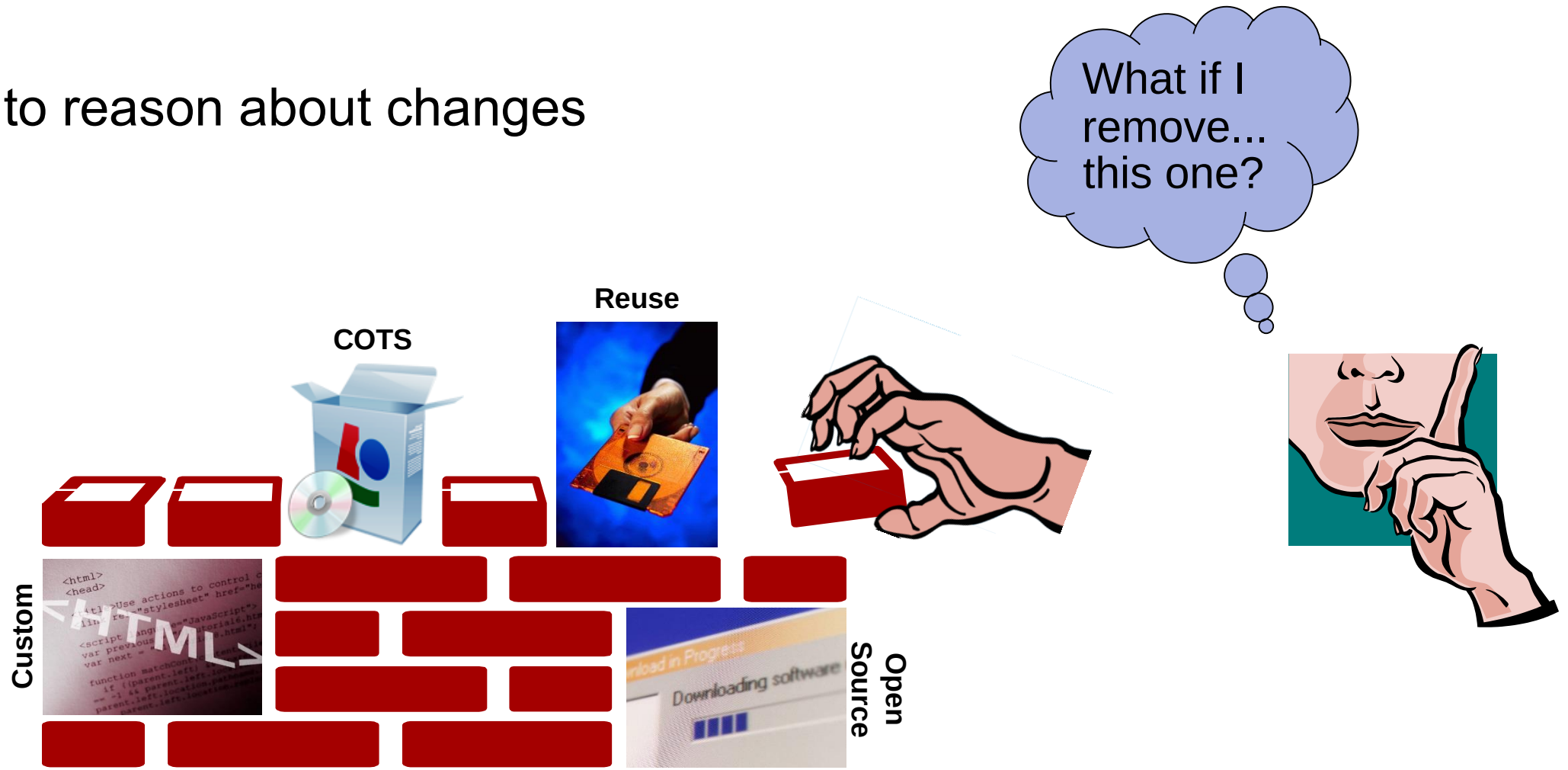


Most modern systems are hybrids

© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

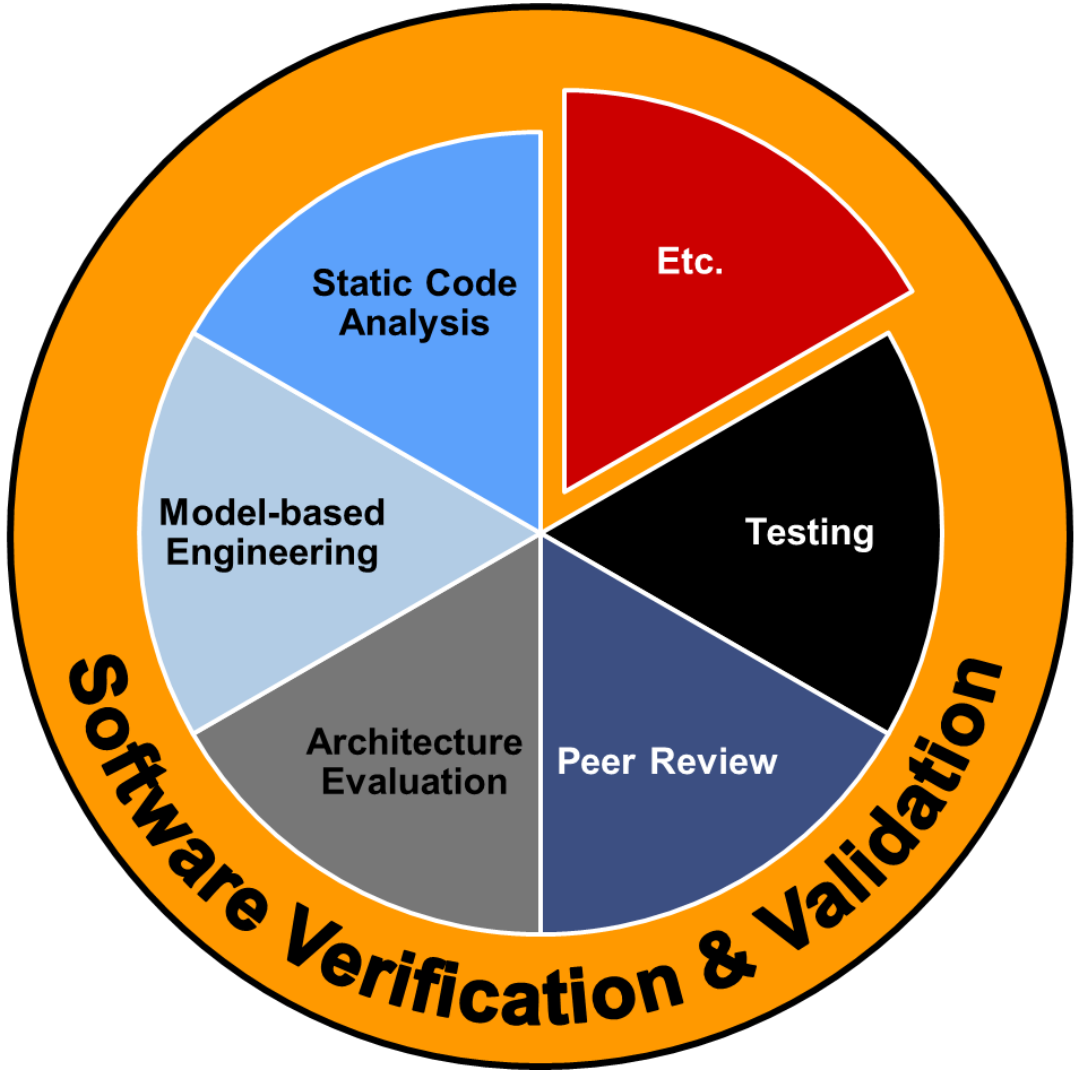
We can employ software architecture to provide a foundation 2 of 2

…and to reason about changes



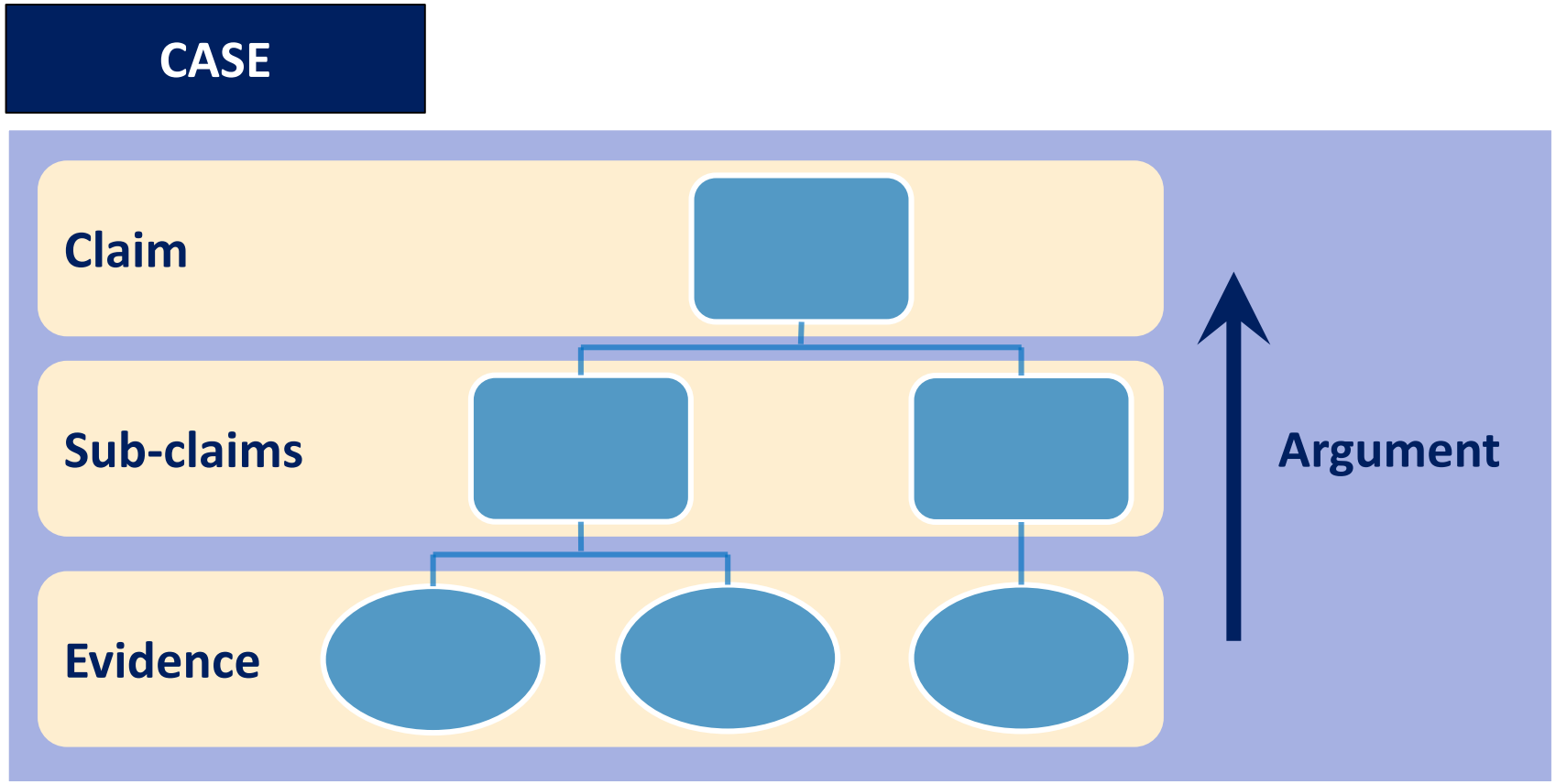
© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

We can take a more holistic view of V&V 1 of 2



© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

We can take a more holistic view of V&V 2 of 2

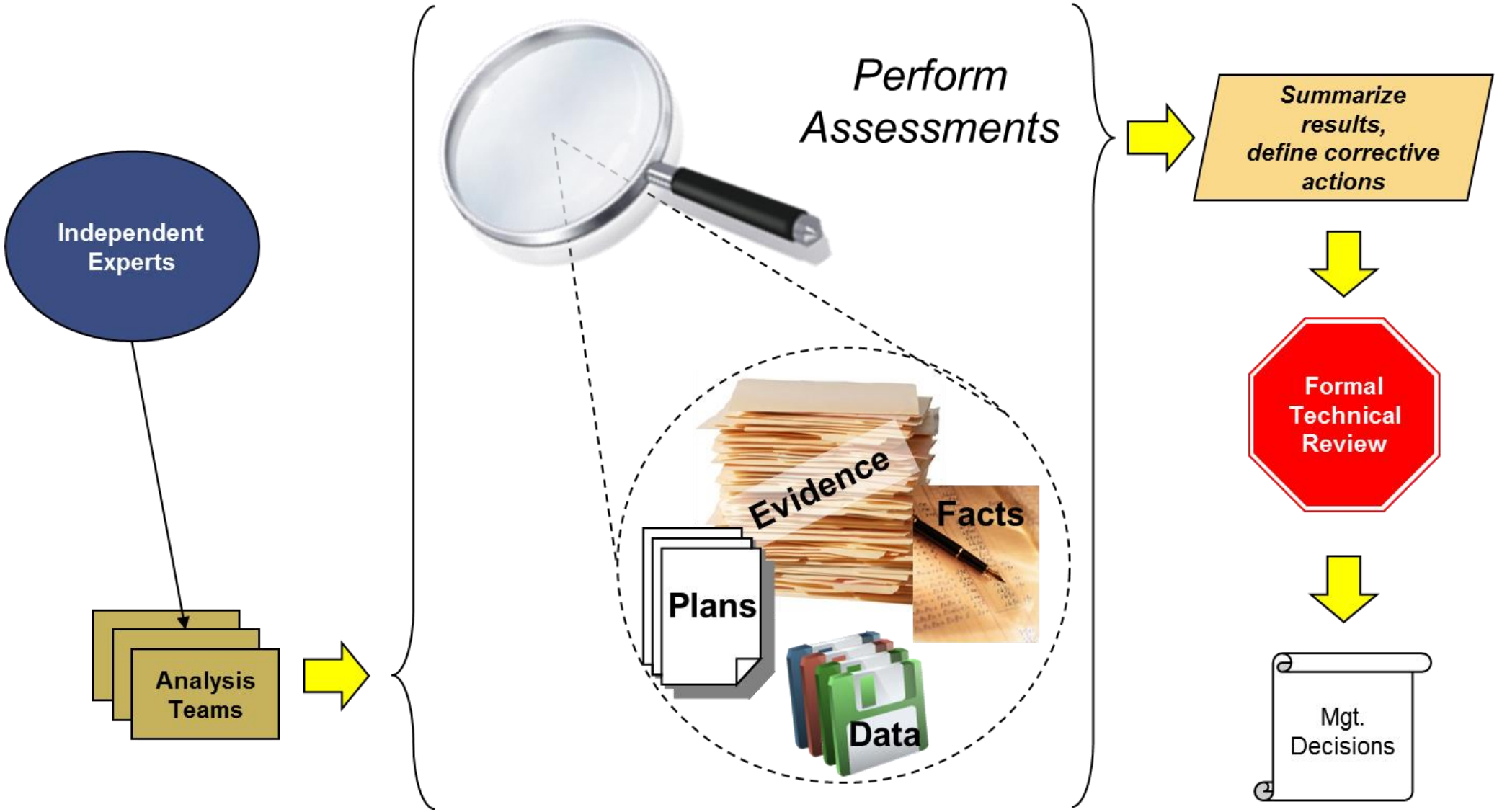


© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

We can employ *evidence-based* reviews

© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

We can employ *evidence-based* reviews

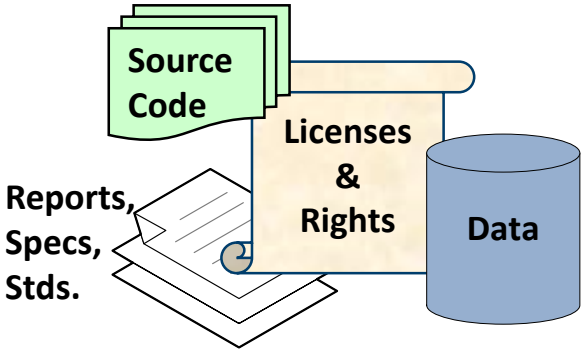


© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Most important – we need to start making changes!

Stop pretending software doesn't matter!

- Include software experts in project planning



Pay attention to long-term sustainment drivers

- Plan for licensing and data rights
- Buy the work products necessary for owning the technical baseline

Stop being afraid of software

- We know there are challenges, so let's face them head on!

Reduced complexity is not in the cards; we need to be better engineers & managers

© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

For More Information

Stephen Blanchette Jr.
Deputy Director, Client Technical Solutions
Chief Engineer, Army Programs
Telephone: +1 703-247-1385
Email: sblanche@sei.cmu.edu

Web
www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

U.S. Mail
Software Engineering Institute
4401 Wilson Boulevard
10th Floor
Arlington, VA 22203-2612
USA

SEI Customer Relations
Email: info@sei.cmu.edu
Telephone: +1 412-268-5800
SEI Fax: +1 412-268-6257

© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.