

# AADL Overview and Perspectives

Peter Feiler

Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

Copyright 2019 Carnegie Mellon 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.

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.

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](mailto:permission@sei.cmu.edu).

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

DM19-1081



## **Safety Critical Embedded Software System Challenge**

**SAE AADL Standard and Virtual System  
Integration to the Rescue**

**Embedded Software System Qualification  
and Assurance**

# The Safety Critical Embedded Software System Challenge

## Problem:

Software increasingly dominates safety and mission critical system development cost.

80% of issues discovered post unit test.

**Solution:** Early discovery of system level issues through virtual Integration and incremental analytical assurance.

## Approach:

International standard based technology matured into practice through pilot projects and industry initiatives.

Open source research prototyping platform continually enhances analysis, verification, and generation capabilities.

***Reduced Defect Leakage through Early Analytical Assurance is Critical***



# We Rely on Software for Safe System Operation

## Quantas Airbus A330-300 Forced to make Emergency Landing - 36 Injured

Written by htbw on Oct-7-08 1:48pm  
From: soyawannaknow.blogspot.com



Thirty-six passengers and crew were in a mid-air drama that forced a Qantas Airbus A330-300 to make an emergency landing, the Australian Civil Aviation Safety Authority said Tuesday.

The terrifying incident saw the Airbus A330-300 make a mayday call when it suddenly changed

from Singapore to Perth, Qantas said.

**Australian Transport Safety Bureau** said yesterday. The plane fell 650 feet within seconds, slamming passengers and crew against the ceiling, before the pilots regained control.

"This appears to be a unique event," the bureau said, adding that Airbus, Toulouse, France-based Airbus, the world's largest maker of commercial aircraft, issued a telex late yesterday to airlines that fly Airbus A330-300s and A350-900s fitted with the same air-data computer. The advisory is "aimed at minimizing the risk in the unlikely event of a similar occurrence."

## Two Crashes In Five Months

## What's Wrong with Boeing's 737 Max 8?

Boeing's new airplane has only been around for two years and already two 737 Max 8s have crashed, killing 346 people. The disasters may be attributable to a design flaw that emerged when engineers began cutting corners.

Boeing's Max 8 is short, limiting ground clearance under the wings. The engine simply doesn't fit.

## FAA says software problem with Boeing 787s could be catastrophic

By Dan Catchpole  
@dcatchpole

The Federal Aviation Administration says a software problem with Boeing 787 Dreamliners could lead to one of the advanced jetliners losing electrical power in flight, which could lead to loss of control.

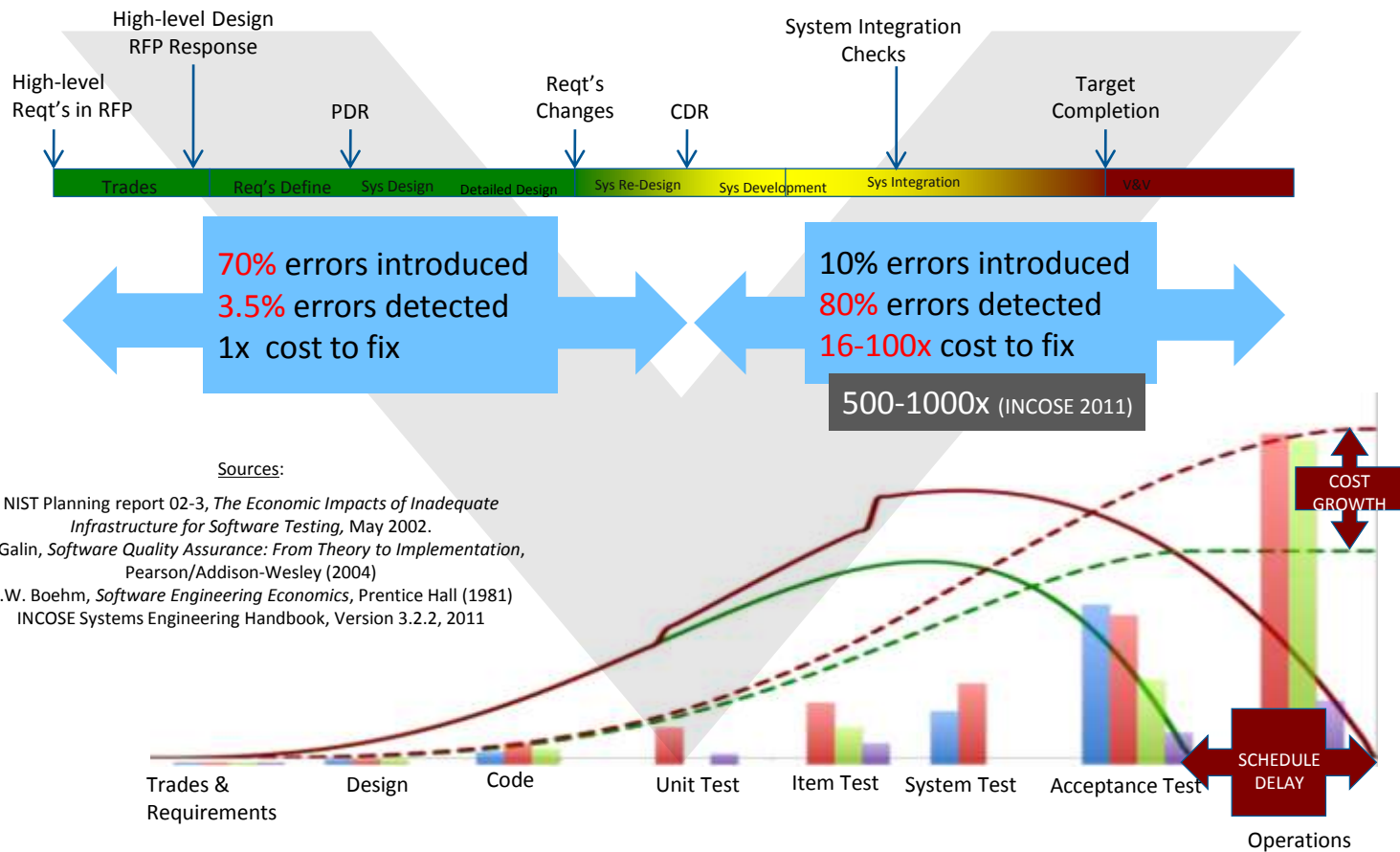
- The Buzz:** Hipster's dilemma
- Boeing & aerospace news
- Aerospace blog

The FAA notified operators of the airplane Friday that if a 787 is powered continuously for 248 days, the plane will automatically shut down its alternating current (AC) electrical power.

Embedded software systems introduce a new class of problems not addressed by traditional system safety analysis

Breakdown in human intensive safety assessment process

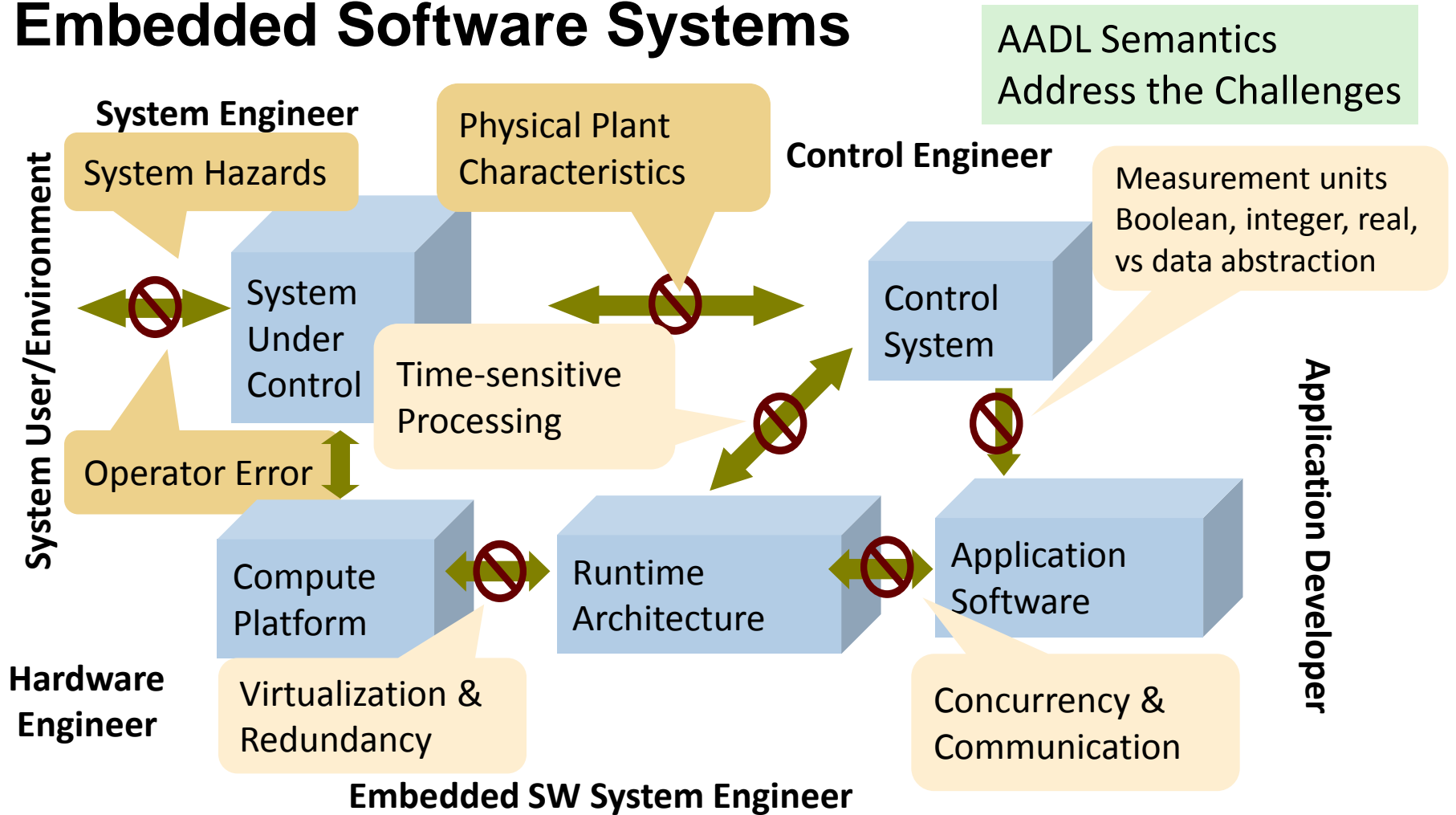
# Current Practice: Impact on Cost and Schedule



Software as % of total system development cost  
 1997: 45% → 2010: 66% → 2024: 88%

Post unit test software rework currently  
 ~50% of total system development cost

# Technical Challenges in Safety-Critical Embedded Software Systems



*Why do system level failures still occur despite best safety practices?*

*Embedded software systems have become a major **safety** and **cyber security** risk*

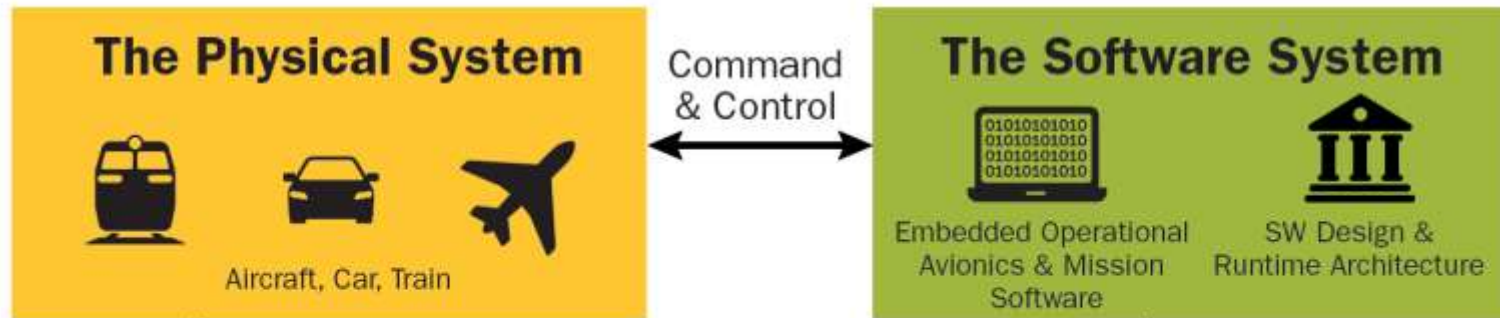
# Safety Critical Embedded Software System Challenge

**SAE AADL Standard and Virtual System Integration to the Rescue**

**Embedded Software System Qualification and Assurance**



# Architecture Analysis & Design Language (AADL) Standard Targets Embedded Software Systems



**SAE International**  
AS 5506 Standard Suite  
Standards provide long-term industry-wide solutions to support multi-organization model-based engineering

In 2008 Aerospace industry initiative chose AADL over SysML and other notations as it specifically addresses embedded software systems

**AADL captures mission and safety critical embedded software system architectures in virtually integrated analyzable models to discover system level problems early and construct implementations from verified models**

# SAE International AADL Standard Suite (AS-5506 series)

## Core AADL language standard [V1 2004, V2 2012, V2.2 2017]

- Focused on embedded software system modeling, analysis, and generation
- Strongly typed language with well-defined semantics for execution of threads, processes on partitions and processor, sampled/queued communication, modes, end to end flows
- Textual and graphical notation
- Revision V3 in progress: interface composition, system configuration, binding, type system unification

### Standardized AADL Annex Extensions

- Error Model language for safety, reliability, security analysis [2006, 2015]
- ARINC653 extension for partitioned architectures [2011, 2015]
- Behavior Specification Language for modes and interaction behavior [2011, 2017]
- Data Modeling extension for interfacing with data models (UML, ASN.1, ...) [2011]
- AADL Runtime System & Code Generation [2006, 2015]

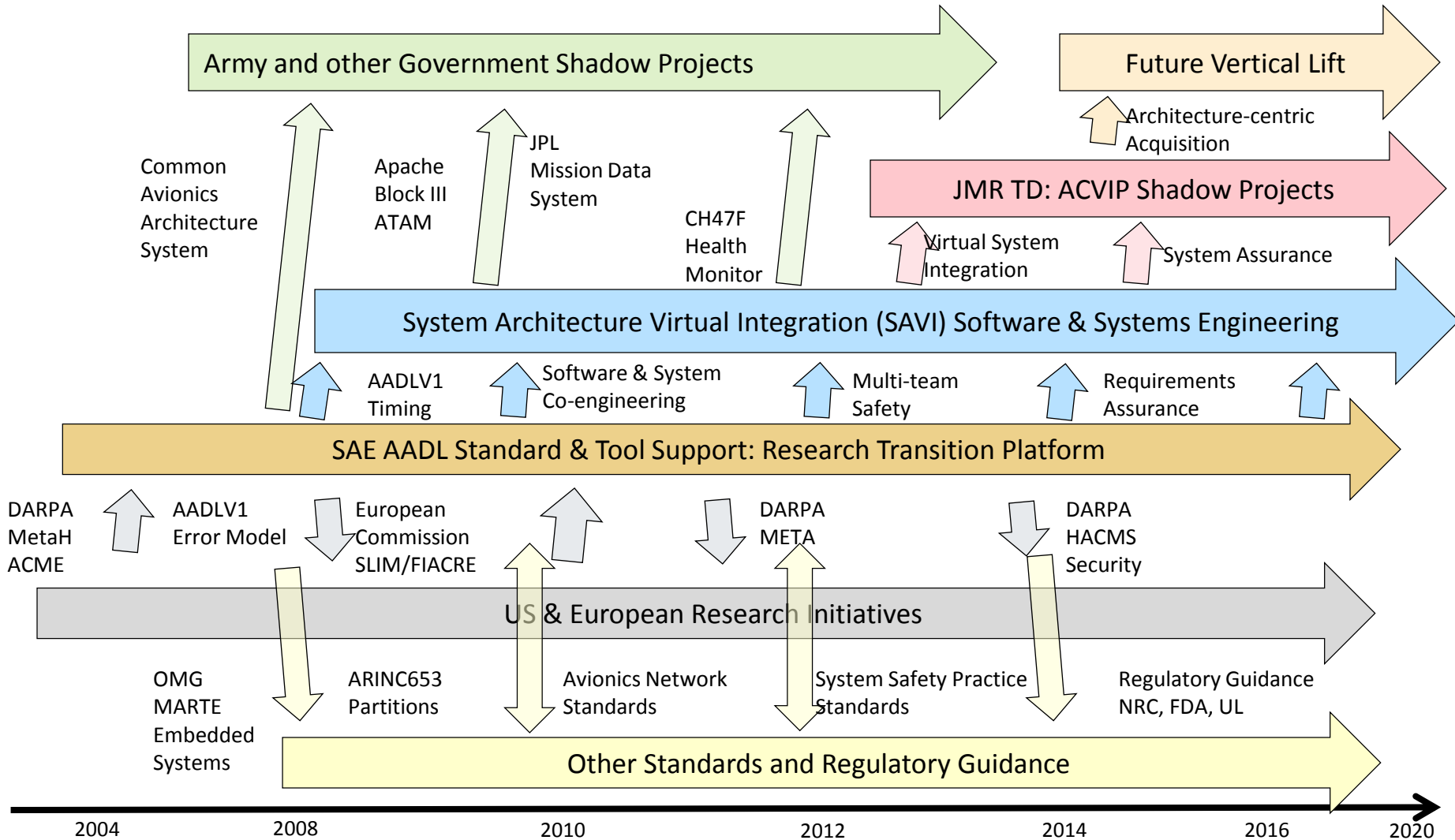
### AADL Annexes in Progress

- Network Specification Annex
- Cyber Security Annex
- FACE Annex
- Requirements Definition and Assurance Annex
- Synchronous System Specification Annex

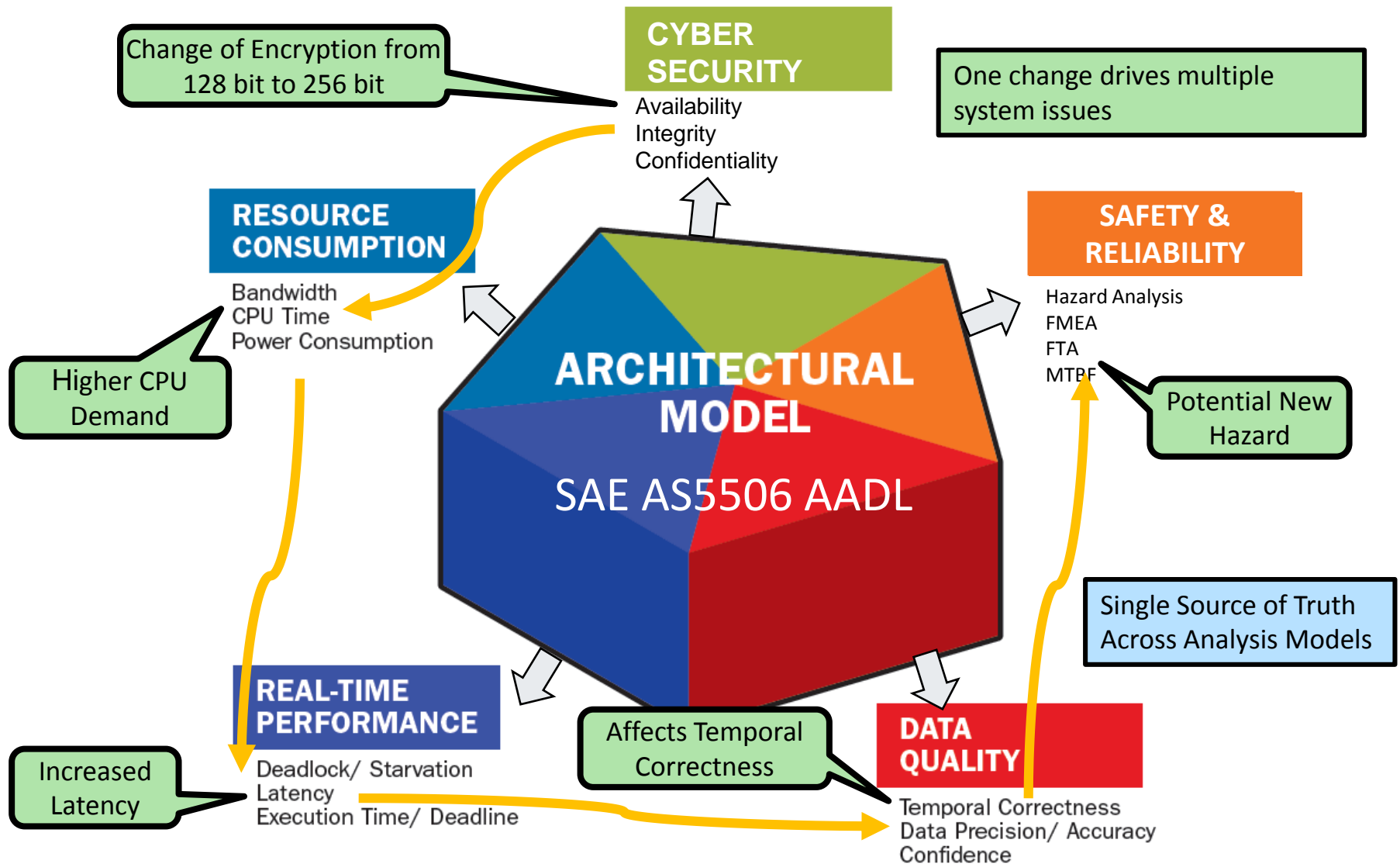
# SAE AADL & Architecture-centric Virtual Integration

## Evolution, Maturation and Transition

**AMRDEC has funded AADL standards development since 1999**



# Analysis of System Properties via Architecture Model A Contribution to Single Source of Truth



# Latency and Jitter Contributors

Control System Engineering View

Processing latency

Sampling latency

Physical signal latency

## Software System Latency Contributors

Execution time variation: algorithm, use of cache

Processor speed

Resource contention

Preemption

Legacy & shared variable communication

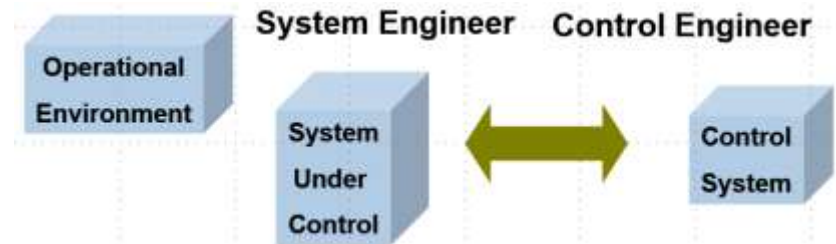
Rate group optimization

Protocol specific communication delay

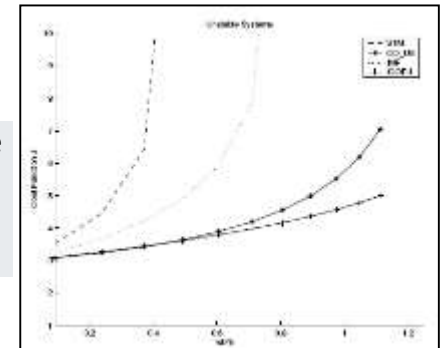
Partitioned architecture

Migration of functionality

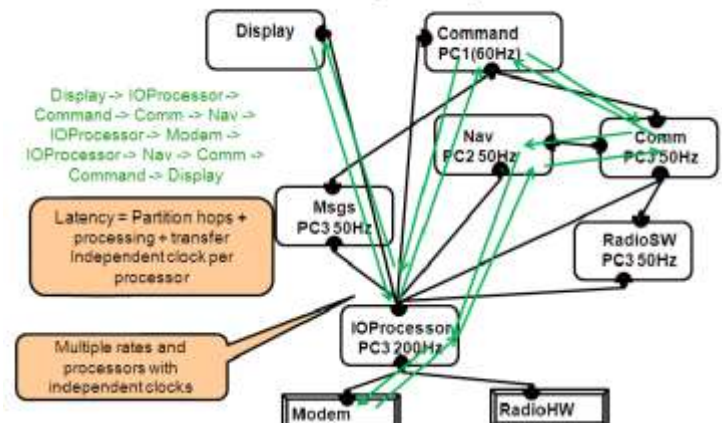
Fault tolerance mechanisms



Impact of Scheduler Choice on Controller Stability  
A. Cervin, Lund U.  
CCACSD 2006



## Flow Use Scenario through Subsystem Architecture



# Safety Critical Embedded Software System Challenge

SAE AADL Standard and Virtual System  
Integration to the Rescue

**Embedded Software System Qualification  
and Assurance**

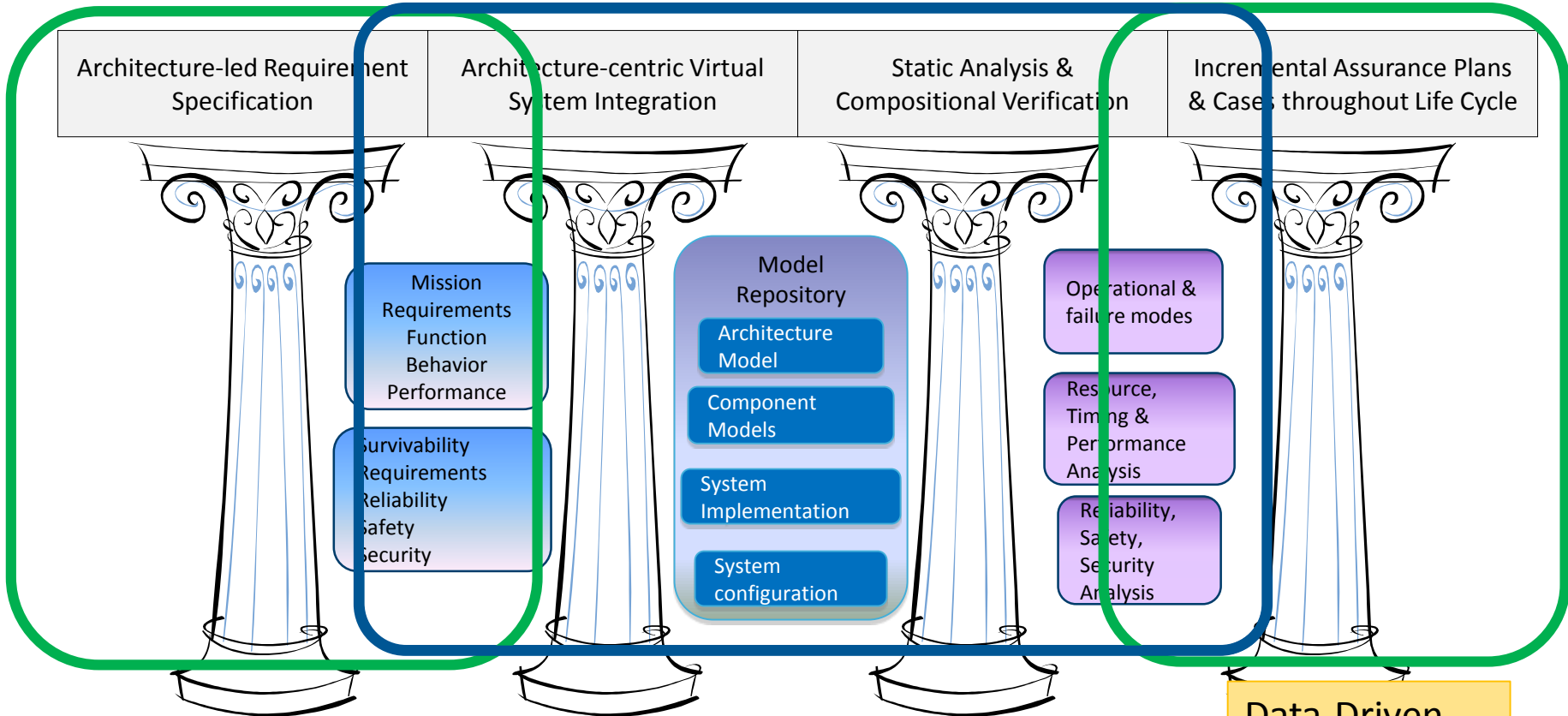


# Assurance & Qualification Improvement Strategy



Assurance: Sufficient evidence that a system implementation meets system requirements

2010 SEI Study for AMRDEC  
Aviation Engineering Directorate

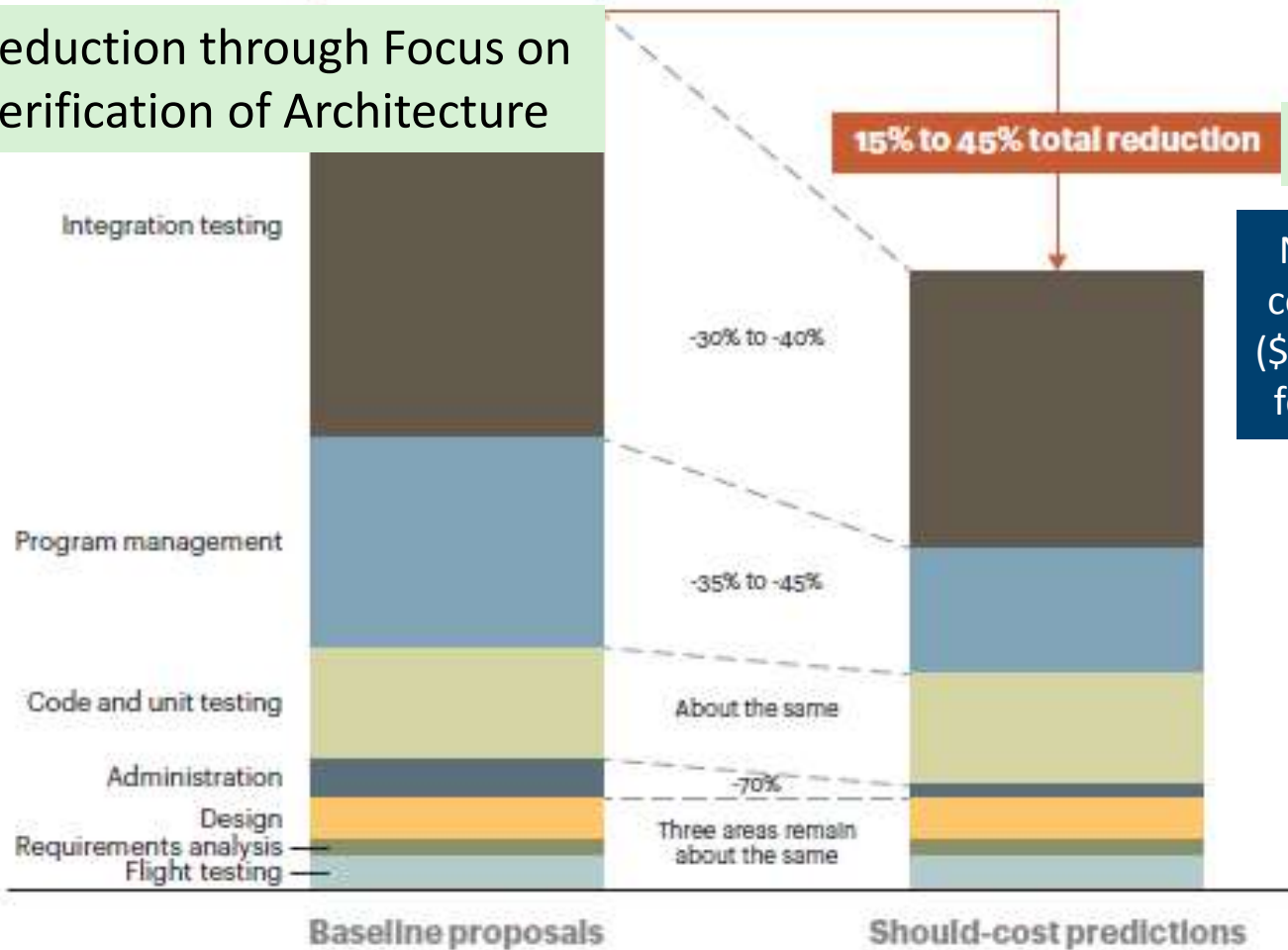


Architecture Centric Virtual System Integration Practice (ACVIP)  
Architecture Led Incremental System Assurance (ALISA)

Data-Driven  
High Leverage  
Cost Effective

# Cost Reduction Potential through Virtual Integration of Embedded Software Systems

Reduction through Focus on Verification of Architecture



ROI on AADL Pilot

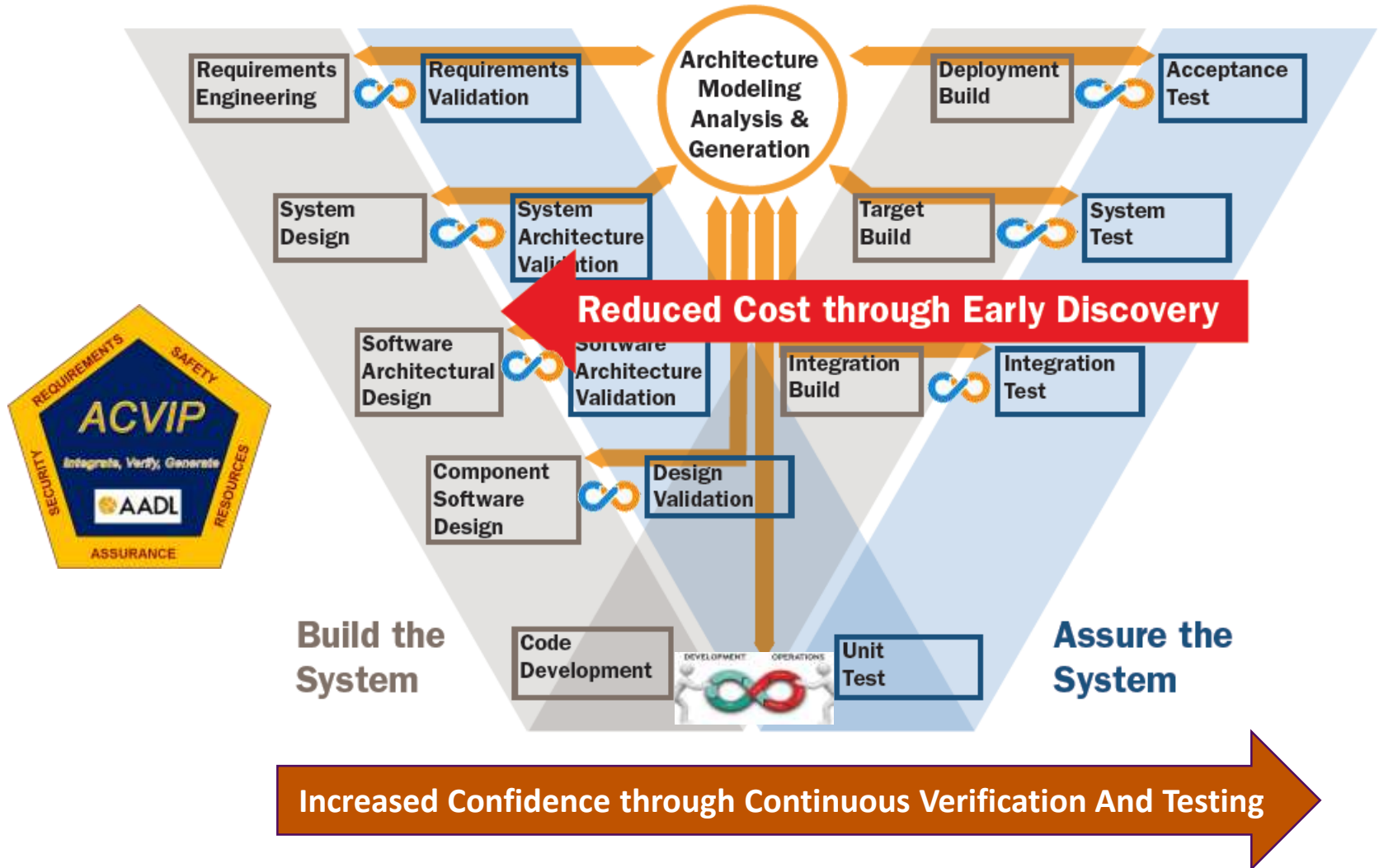
Nominal development cost reduction of 26.1% (\$2.391B out of \$9.186B) for a 27 MSLOC system

Source: *ROI Analysis of the System Architecture Virtual Integration Initiative*  
CMU/SEI-2018-TR-002



ATKearney "Software: The Brains Behind U.S. Defense Systems"

# Benefits of Virtual System Integration & Continuous Lifecycle Assurance



# Summary

Safety Critical Embedded Software Systems are facing exponential growth in software development cost exceeding 70% of total system development cost

AADL is basis for a set of technologies and practices that specifically have been designed to provide early detection and continuous verification throughout the life cycle