

# AWD Control Architecture for Rapid Application Deployment

David Bruder, PE  
Chief Engineer, Systems Engineering

Dan Reich  
Manager, Controls

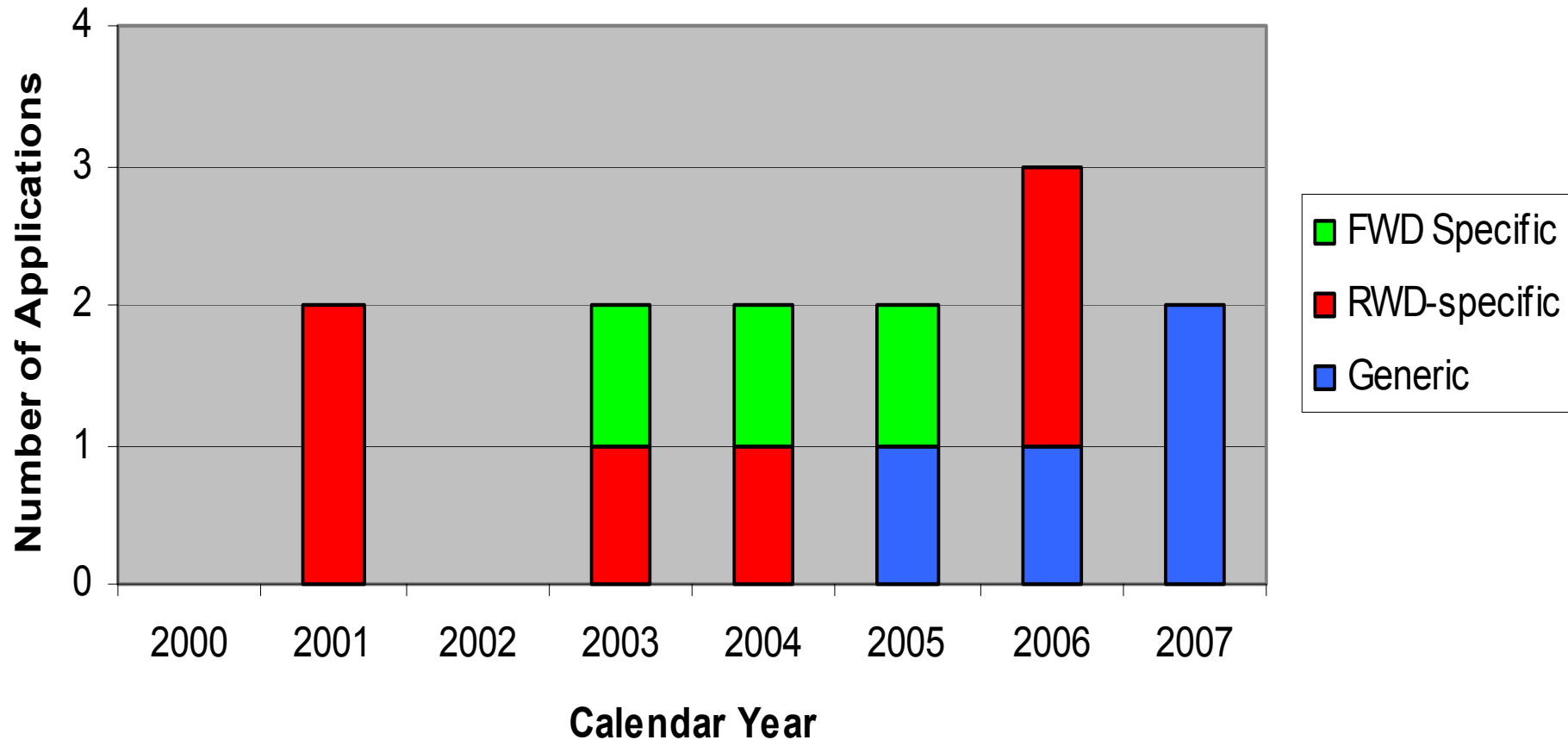
# Introduction

- Need / Background
- BW controls architecture description
- Test cases
- Module reuse

# Background

- Increasing number of active applications
- Increasing requirements
- Need for adaptation to different actuators and drivelines
- Need for improved engineering productivity

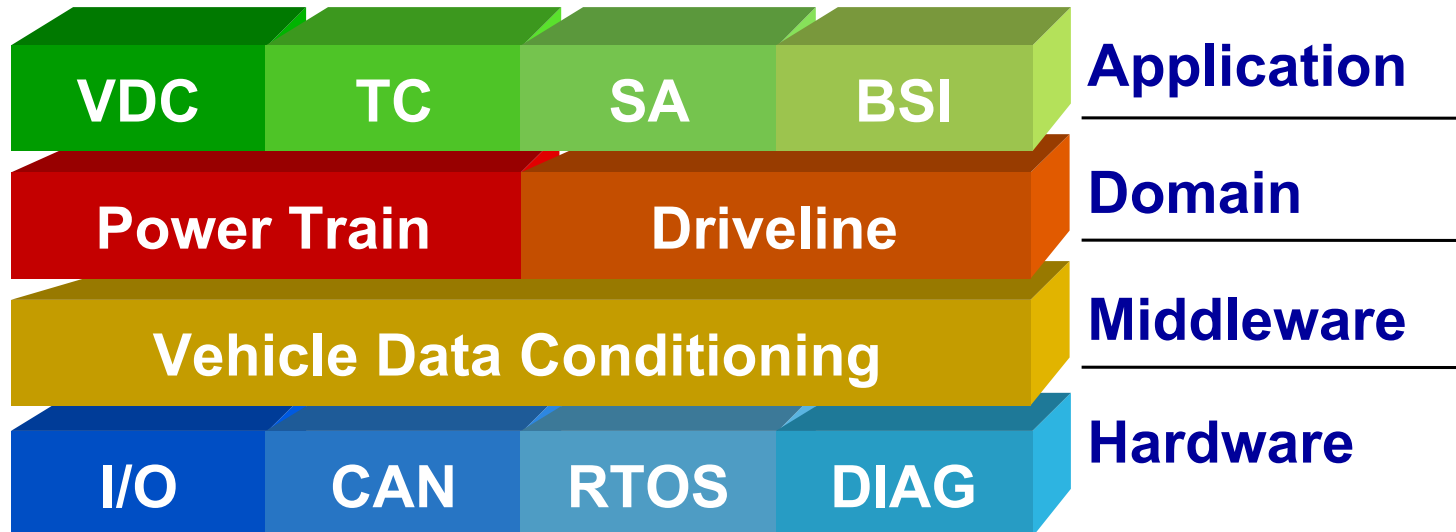
# BorgWarner Active Coupling Control



# Design Objectives

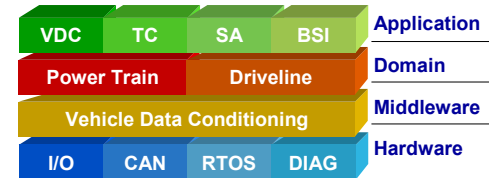
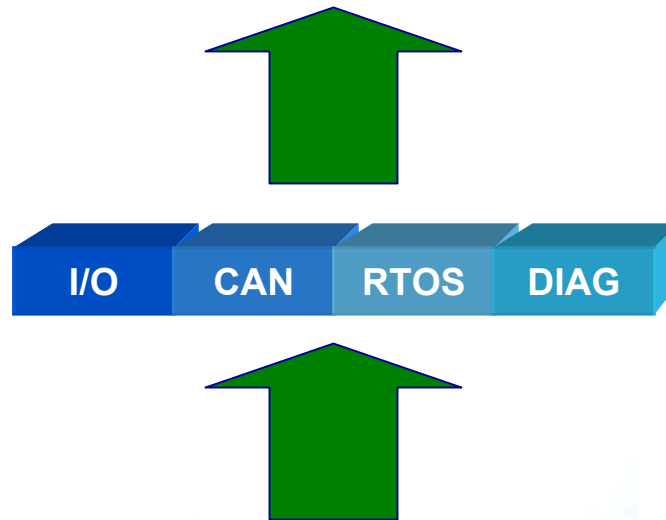
- Stable, Reusable Architecture
- Module Reuse and Decoupling
- Decoupled Actuator and Vehicle Control
- Multiple Actuators (Scalable)
- Easily Adaptable to Different Drivelines
- Physical Parameter Based

# Controls Architecture



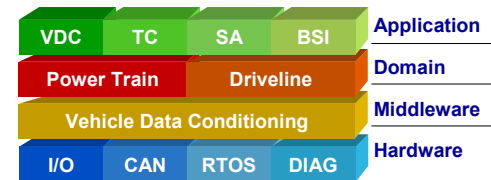
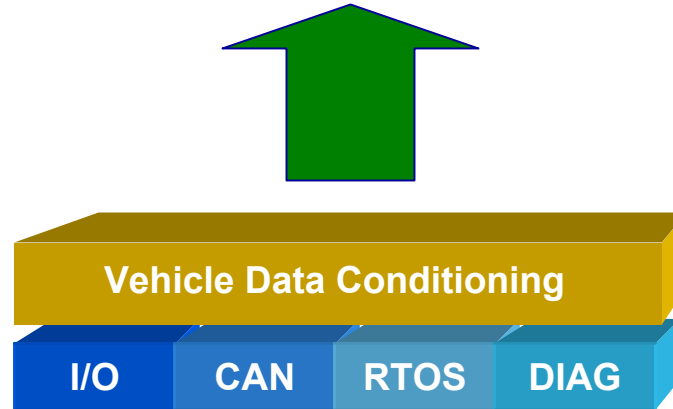
# Core Controls Software

Raw Sensor Data



# Core Controls Software

## Filtered Data in Proper Units

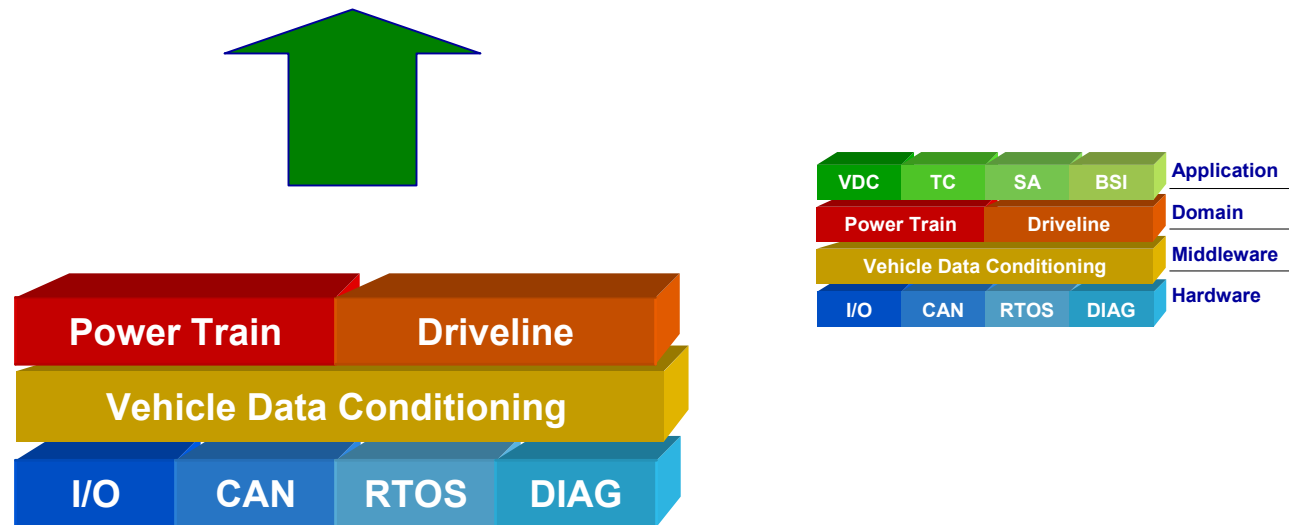


Importance:

- Allows Domain & Application Layers to be unchanged despite any changes to the electronics hardware

# Core Controls Software

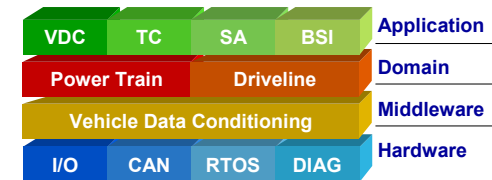
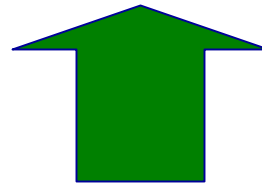
## Scaled, Filtered Data plus Modeled Parameters



Importance:

- This layer is tailored to an application using vehicle-specific, measurable parameters

# Core Controls Software

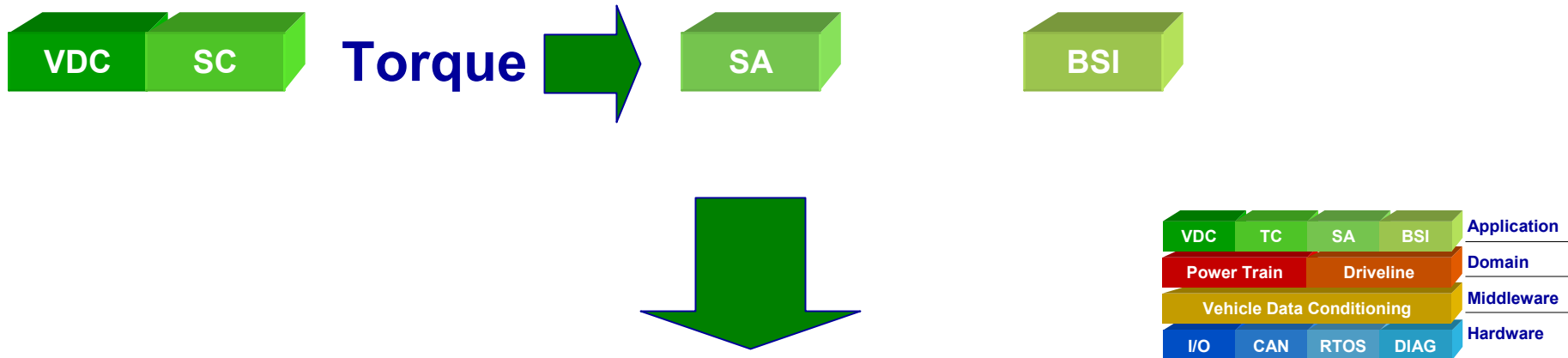


## Scaled, Filtered Data plus Modeled Parameters

Importance:

- This layer is vehicle-, driveline-, and OEM-independent
- Vehicle control is tailored to an application using tunable parameters
- Actuator control is specific to actuator type and is tailored using measurable parameters

# Core Controls Software

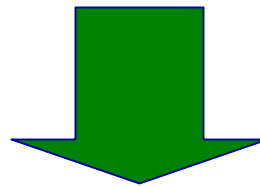


## Actuator Command

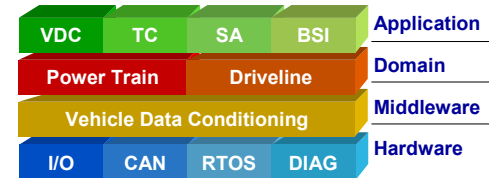
Importance:

- Separation of vehicle-level control from actuator-specific control
- Hardware changes do not affect the vehicle level control
- Changes to the vehicle do not affect the actuator control

# Core Controls Software

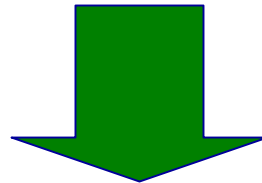
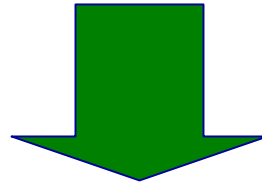


**Actuator Command**

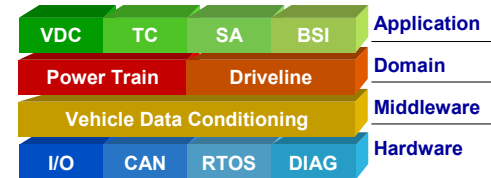


# Core Controls Software

Actuator Command

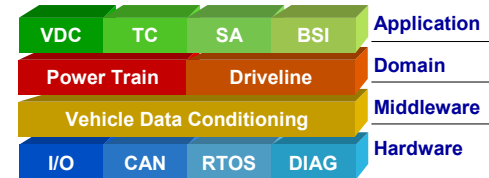
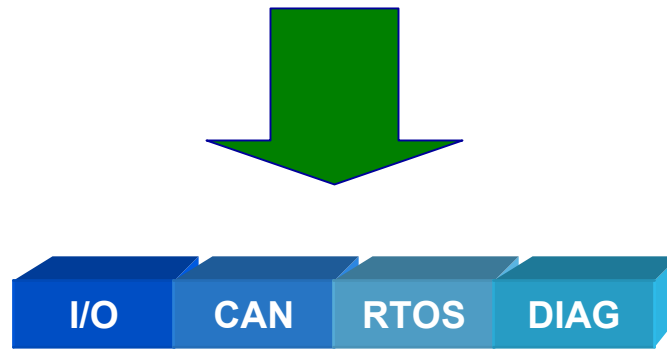


Hardware Specific Command



# Core Controls Software

## Hardware Specific Command



Output Control

# Architecture Effectiveness

- Case 1: Different Actuators, Same Vehicle
  - Goal: Change Smart Actuator only
- Case 2: Different Vehicles, Same Actuator
  - Goal: No tuning change

# Case 1: Identical Vehicles with Different Coupling Technologies



ITM 3e™  
(electromagnetic)

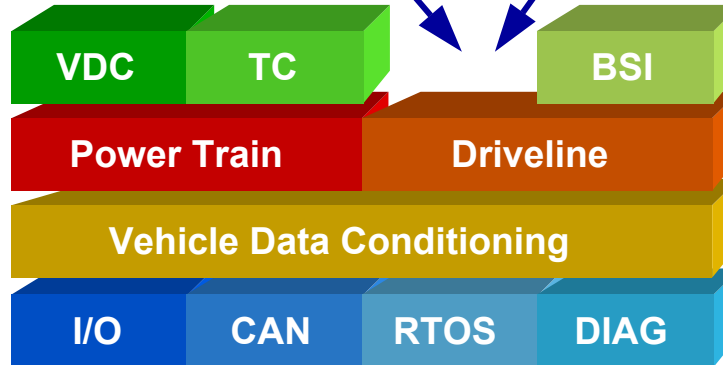


Audi A3

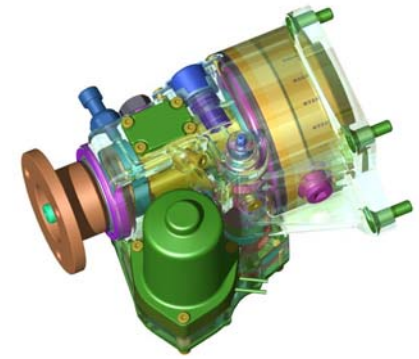
ITM 3e Smart Actuator



PTM Smart Actuator



Carryover controls and vehicle tuning



PTM™  
(electro-motor)



Audi A3

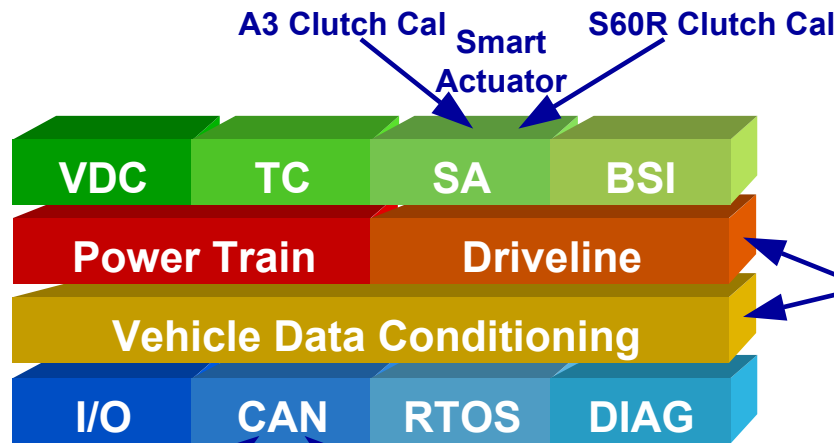
# Case 2: Different Vehicles with Identical Coupling Technologies



ITM 3e™  
(electromagnetic)



Audi A3



A3 database

S60R database

Carryover controls and vehicle tuning



ITM 3e™  
(electromagnetic)



Volvo S60R

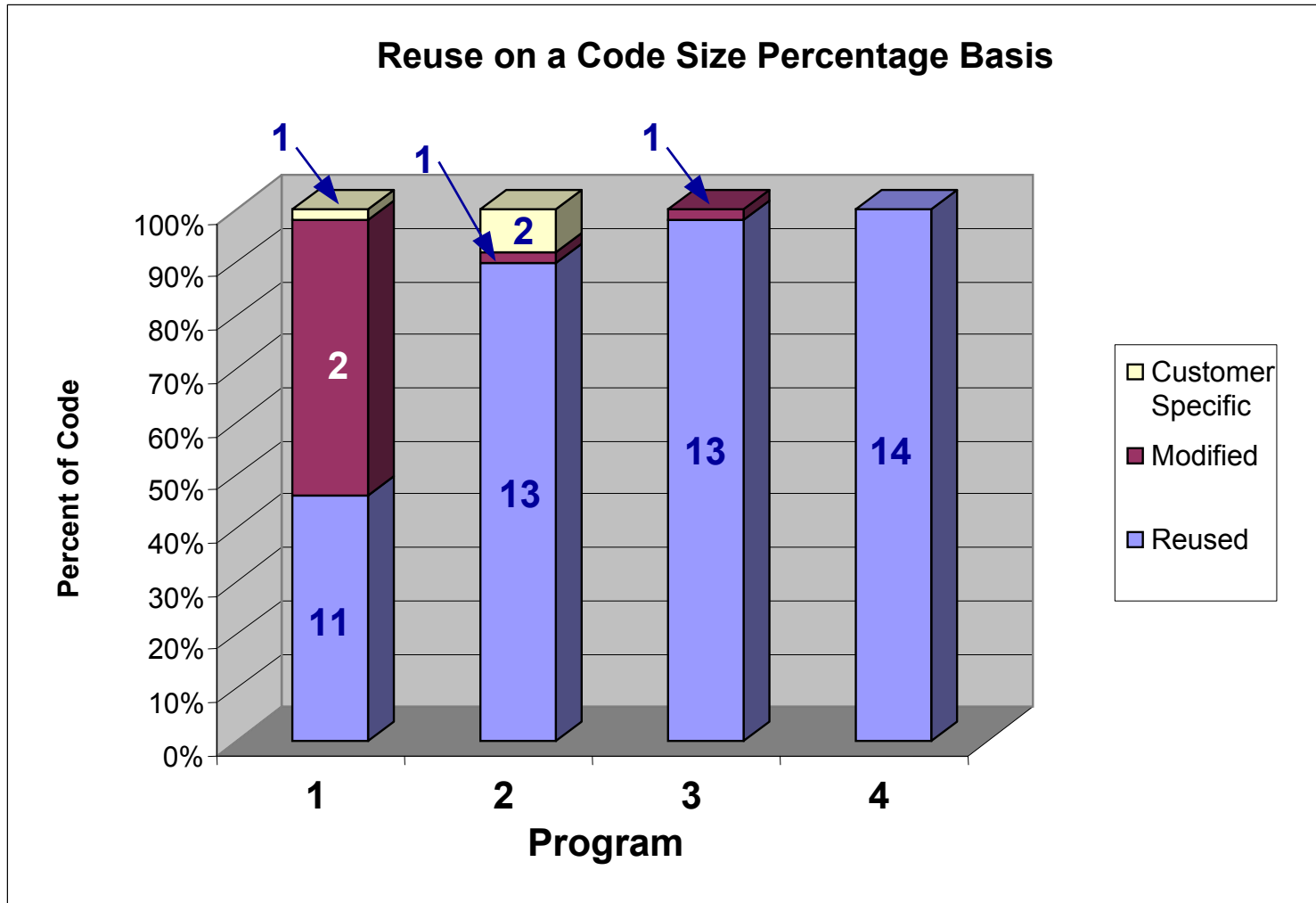
# Module Reuse

- Module reuse – use whole module intact
  - Any small change means “modified”
- Applications contain 14-16 modules
- Middleware, Domain, Application Layers

# Module Reuse

Program	Modules Reused Completely	Modules Reused w/ Modification	New Modules Customer Specific	Total Modules
FWD AWD 1	11	2	1	14
FWD AWD 2	13	1	2	16
RWD AWD 1	13	1	0	14
RWD AWD 2	14	0	0	14

# Module Reuse



# Conclusions

- Effective reuse of core modules
- Reduced application effort
- Reduced effort to port new hardware
- Enables modular development

**Thank you**