Model-Based Design Approach for Software Architectures in Automotive

Domenico Ferrari
Application Engineer
dferrari@mathworks.com

About me…

Application Engineer at MathWorks for Automotive.

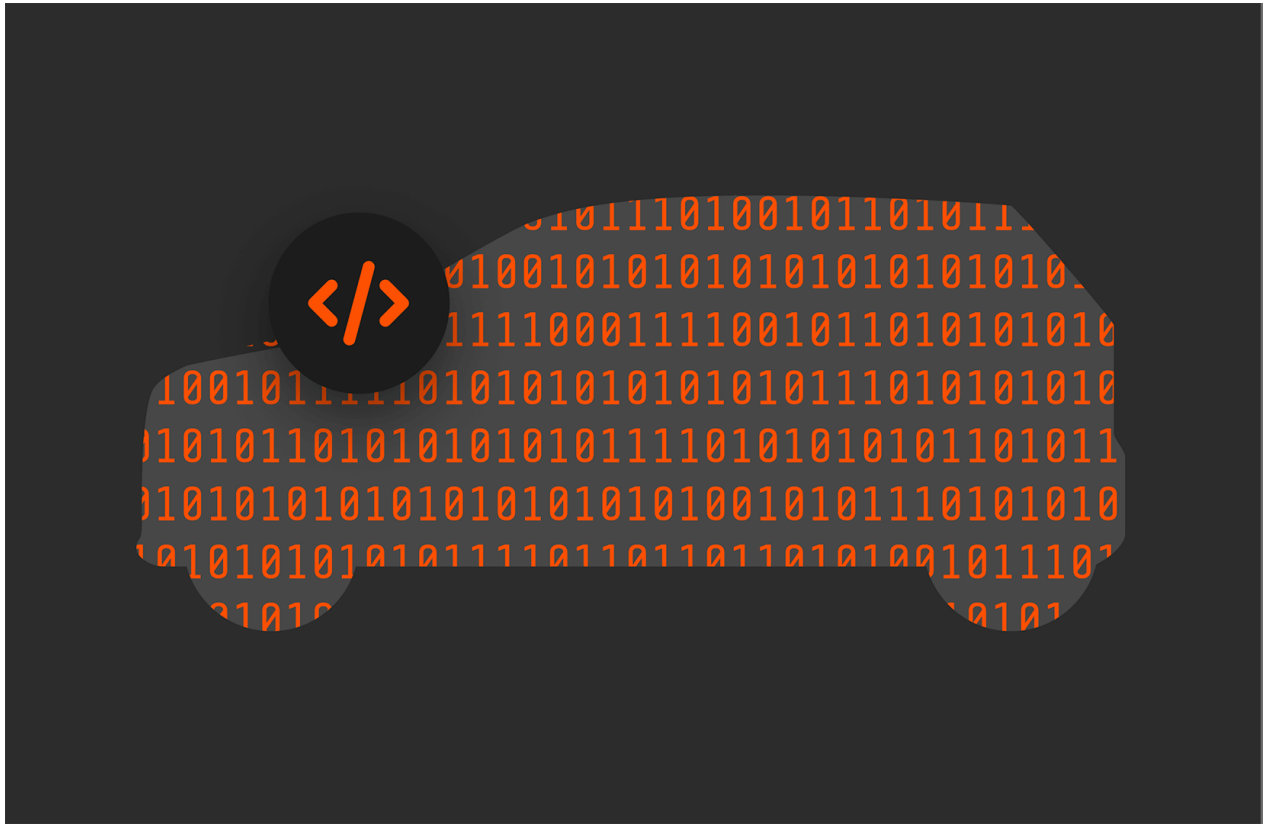
Master Degree in Computer Engineering from the Politecnico di Torino.

Worked in General Motors as Software Engineer, developing algorithms for Diesel Control Engines and in Dumarey as responsible of AUTOSAR methodology.

How do we manage the growing complexity of software?

Quiz: What is the average number of lines of code in a modern car?

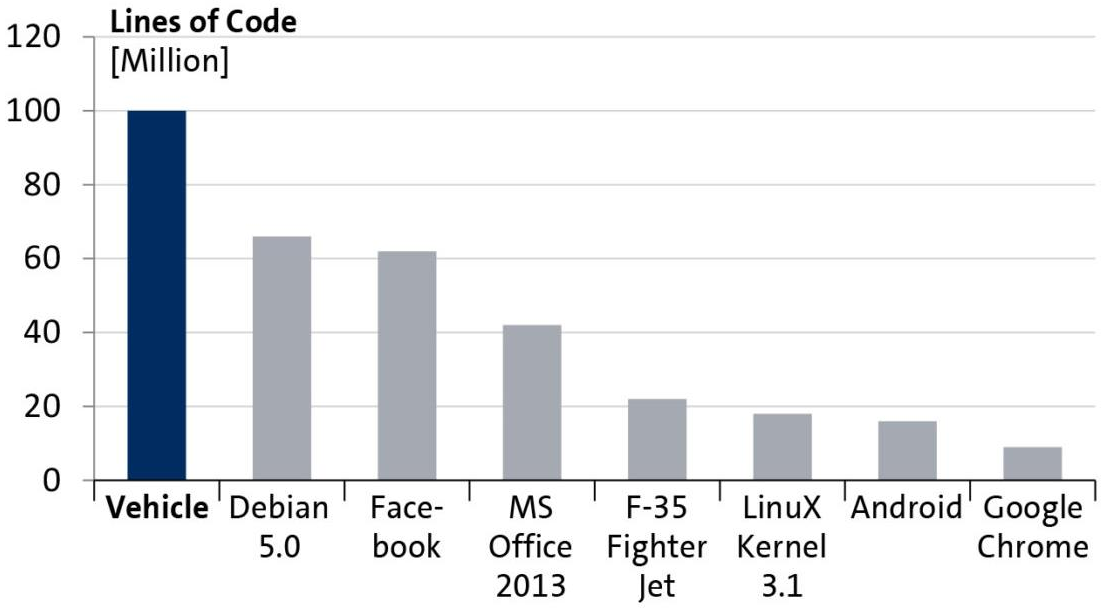
A. 1M
B. 10M
C. 100M
D. 1B



In the future, software will be a main differentiator in the automotive industry…

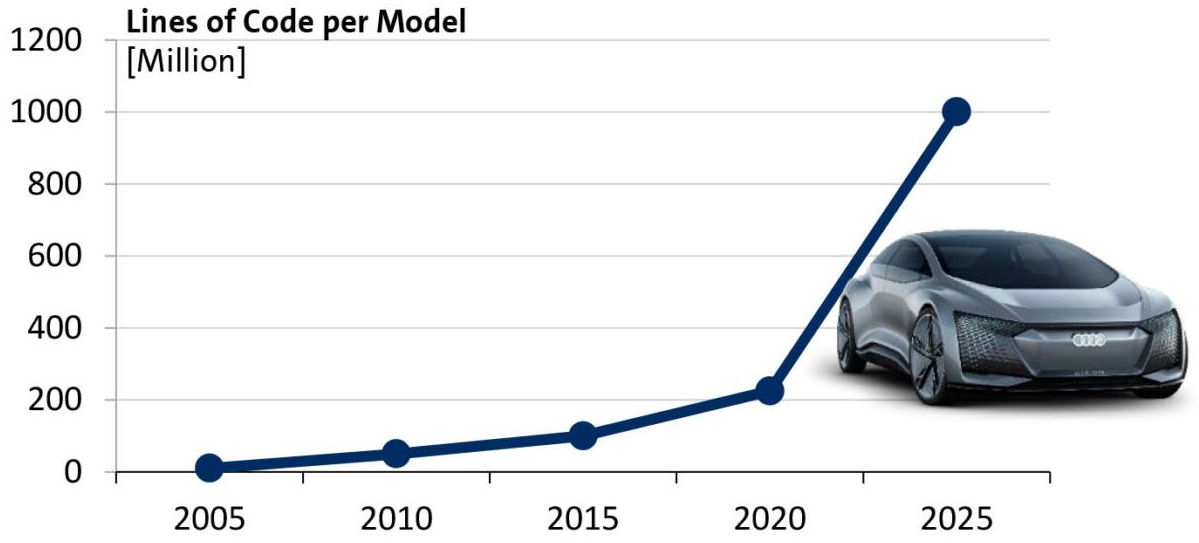
Today

- 100 million lines of code per vehicle
- Approximately $ 10 per line of code
- Example: Navi system 20 million lines of code



Tomorrow (autonomous cars)

- > 200 - 300 million lines of code are expected
- Level 5 autonomous driving will take up to 1 billion lines of code



Sources: https://spectrum.ieee.org/transportation/systems/this-car-runs-on-code | http://frost.com/prod/servlet//press-release.pag?docid=284456381 | https://www.visualcapitalist.com/millions-lines-of-code/

A-SPICE

Automotive Software Process Improvement and Capability dEtermination

- A-SPICE is a framework that defines best practices for embedded software in automotive development.
- It allows teams to organize their projects and approaches to ensure software quality, manageability and reliability.

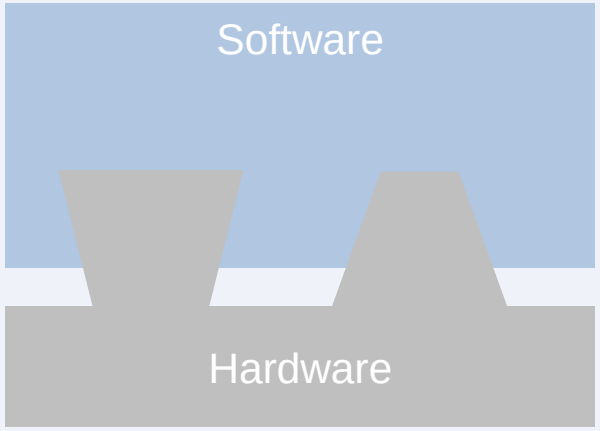
ISO 26262

- ISO 26262 is an international standard for the functional safety of electrical and electronic systems in road vehicles.
- The target of the standard is to identify the safety hazards and mitigate them. Safety risks must be identified and managed at the early stages of product development.

AUTOSAR

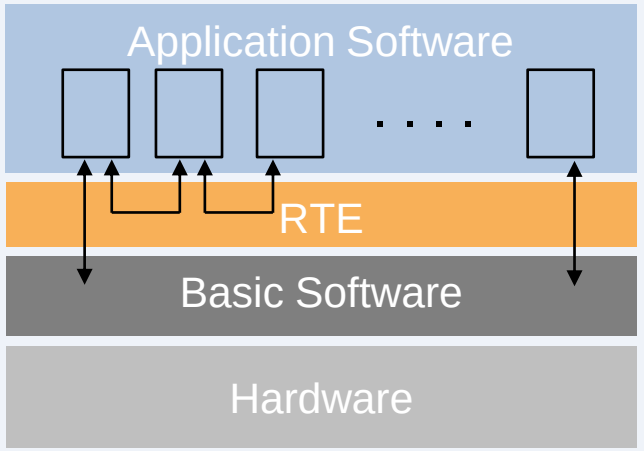
AUTOSAR (AUTomotive Open System ARchitecture) is a global development partnership of car manufacturers to establish an open and standardized software architecture for automotive electronic control units (ECUs).

Non-AUTOSAR



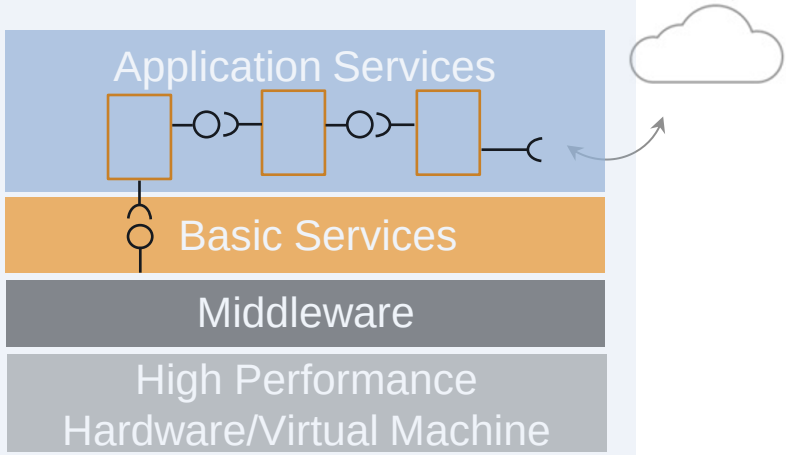
- High SW-HW coupling
- No standard APIs
- No/minimal SW reuse

AUTOSAR Classic



- Static SW component allocation (design-time)
- Signal-based communication
- HW abstraction
- Monolithic update (full image flashing)

AUTOSAR Adaptive



- Dynamic Service discovery (run-time)
- Service-oriented communication
- Higher HW abstraction
- Selective updates (OTA)