Systems Engineering and Security Challenges

Frank Kirschke-Biller
Manager Global Core Software Process

12.10.2017
1. Introduction
2. Current system engineering solutions
3. Future system engineering needs and enhancements
4. Summary and outlook
THE MOBILITY EXPERIENCE.....

"Opening The Highways to all Mankind"

EE Hardware: Magneto & eventually electric Lights
Software: Huh?

"The car is evolving from the ultimate industrial product into the ultimate technology product."
Mark Fields – former CEO Ford Motor Company

EE Hardware: 70+ ECUs, Miles of Cabling
Software: More than 100M Lines of code

Within a more complex and diversified mobility industry landscape, incumbent players will be forced to simultaneously compete on multiple fronts and cooperate with competitors
Automotive revolution – perspective towards 2030 (McKinsey Study)

EE Hardware: To Enable Software
Software: Mobility Ecosystem Enabled Through Software
Features evolved from component functionality to complex systems

Single ECUs

Systems

Complex, Networked Systems

2000

2010+

EXAMPLE: FRONT LIGHTING

Simple Lights (1995)

Adaptive Front Lighting (2005)

Glare Free High Beam System (2015)
Functional growth is going to continue and will even accelerate – software as key innovation driver

- Feature growth mainly driven by IVI and driver assistance
- Distributed feature support move to smart features.

Enabling:
- better customer experience, e.g.
  - aligned and coordinated feature behaviour
  - DNA support (e.g. SYNC)
- more efficient solutions e.g.
  - shared usage of ECUs (e.g. Camera System)
  - modular design for scalable and re-usable systems
  - features as pure software solution (e.g. MyKey)
Agenda

1. Introduction
2. Current system engineering solutions
3. Future system engineering needs and enhancements
4. Summary and outlook
System Engineering initiatives to address challenges

System engineering initiatives

Feature Ownership
- Establish Feature Ownership as true Systems Engineering Role with defined deliverables
- Migration from ECU oriented development to Systems Engineering

Requirements Engineering
- Aligned Document structure
- Specification templates
- Quality Assurance

Model Based System Engineering and Integrated tool chain
- Model Based System Engineering to manage complexity (e.g. SysML)
- PLM and ALM solution to manage data and offer workflows
  - Growing focus on complete system incl. Software and Mechatronics
Using Systems Engineering principles

This is top down, requirement driven design

Contrasts with bottom up, functionality driven by components constraints

Implement consistent Feature & Function Engineering

Feature Ownership = Responsibility across ECUs as well as organizations

Feature/Function Focused Systems Engineering

Introduce dedicated Feature/Function Owner role in PD

Assign one (lead) person per Feature or Function

Core Roles ...

... and Application Roles required
Why is Requirements Engineering (RE) Important?

• Three major product development (PD) project risks related to RE:
  • Missing requirements
  • Wrong requirements
  • Changing requirements

• Any of the risks above lead to major problems of projects, e.g. increased costs, unplanned effort, bad quality, delays and deviation from committed milestones

• RE is to minimize and control these risks (Incl. Software Security!)
  • Goal of RE: have good enough (not perfect) requirements to execute the project with acceptable risk
  • However, there is no single best standard RE process or method that fits all organizations. The RE-Process has to be adopted to fit Ford’s specific needs.
  • Common abstraction levels (move from ECU/component level to Feature/Function level)
  • Common templates (specification content, RE meta data, use cases, scenarios …)
  • Common RE tool and RE review process
Solution: Requirements engineering for software enabled, feature design

Requirements Engineering and Product Architecture interact and go hand in hand.
Solution: Common abstraction levels supported by Requirements engineering (RE) templates

• RE initiative defines template(s) for each abstraction level to master complexity
• Enables consistent RE processes and methods across domains

- Feature Level Requirements
  - Feature 1
  - Feature 2
  - Feature 3

- Function Level Requirements
  - Fnc1
  - Fnc2
  - Fnc3
  - Fnc4

- Component Level Requirements
  - F1
  - F2
  - F3
  - F4

- SW Requirements
  - SW

- HW Requirements
  - HW

- ECU Engineering Specification (per ECU)

- Interface Spec
  - Interface Spec
  - Interface Spec

- Feature Implementation Spec (per E/E platform)
- ECU Functional Spec (per component)
- Ext. Interface Spec
- EE-Platform specific Concepts

- Feature Document (per feature)
- Aggregated Feature Specification (AFS)
- Product Requirements Document (optional)
- Stakeholder Template

• Requirements need interface to architecture specification on each level
Drivers for change: connected vehicle & smart mobility require different system engineering approach
Drivers for change: Examples for connected service features

**E-Call**
Mandatory in-vehicle system that calls emergency services if an automatic eCall event is triggered by detection of a serious road accident or a manual eCall event is triggered by the user.

**Vehicle Status**
Displays the vehicle’s status (at keyoff) on a users smartphone via the embedded modem for e.g. Fuel Level, Oil Life, Tire Pressure, 12V Battery, Odometer, Alarm Activated

**Vehicle Locator**
Display the vehicle’s location on a map in the One Ford Owner App on the user’s smartphone using GPS location sent via embedded modem.

**Door Lock/ Unlock**
Ability to Remotely Lock / Remotely Unlock their vehicle from virtually anywhere in the world, using the One Ford Owner App on their smartphone and an embedded modem

**WI-FI Hot Spot**
Wi-Fi Hotspot in the vehicle for passengers to connect Wi-Fi capable devices (e.g. iPad, computer, or cell phone ) and stream internet data through the cellular connection provided by the vehicle’s embedded modem

**Online Traffic**
Navigation systems to include re-routing or or regularly updated indication of the delay caused by traffic jams etc based on up-to-date traffic information

**Ford Proprietary - 10/10/2017**
Key implications from a Systems Engineering perspective

1 Complexity of advanced distributed features will further increase with growth/shift from on-board to off-board

- Current skill set of development engineer / Feature Owner maybe not sufficient
- IT competence for E2E solution and Security required
- Competency and organizational challenges – growing interfaces
- Performance depends on external factors (off-board performance), not only on vehicle design
- Organizational interfaces become more complex
- Feature Specification includes vehicle and “cloud” functional requirements
- Feature Owner must coordinate both vehicle and off board development efforts
Vehicle Architecture (HW and SW) is changing significantly

- External interfaces e.g. embedded modem
- Increase of communication bandwidth, e.g. using Ethernet
- New Design pattern (e.g. Service based communication)
- Extended infrastructure for over the air SW download
- Move from Microcontroller to Microprocessors
- Different functional deployment (Domain/Zone oriented) → more flexible SW deployment
- Alternative operating systems (Unix, AUTOSAR Adaptive etc.)
Key Implication - Cybersecurity

Vehicle Architecture (e.g. Firewall, Encrypted communication) plays major role.

- Security Process utilizing best practice and (future) standards
  - SAE J3061
  - ISO-SAE AWI 21434 Road Vehicles – Cybersecurity Engineering
  - AUTOSAR
  - AUTO ISAC

Hackers are interested in:

- **Money**
  - through manipulation of:
    - Mileage counter/ Truck tachograph
    - Airbag controller
    - Headunit
    - Engine/transmission controllers
    - Plagiarism

- **Scientific Reputation**
  - by proving the:
    - Threat potential
    - Weakness of todays security

- **Extortion**
  - by threatening:
    - OEMs
    - Countries
    - ... to immobilize traffic
    - ... to destruct image
1. Introduction
2. Current system engineering solutions
3. Future system engineering needs and enhancements
4. Summary and outlook
Combined Methodologies & Standards will be required for Effective & Timely development.

- Relationship of Cyber Security, Functional Safety, SW Quality & overall SW process to be considered.
DEVELOP A CYBERSECURITY CULTURE

- Set up a dedicated cybersecurity team and process
- Develop training and awareness programs

VEHICLE CYBERSECURITY LIFECYCLE

- Define information security management system
- Provide security by design / architecture
- Implement security functions, e.g.
  - Security logs
  - Communication protection
  - Control keys and access
  - User data protection
  - Identification, authentication, authorization
  - Self-protection

COLLABORATION IN INDUSTRY

- Improve information sharing amongst industry actors
- Joint work and standards and best practices

VEHICLE CYBERSECURITY LIFECYCLE

- Securing the supply chain, incl. SW QA
- Implement appropriate testing for security functions
- Self auditing and testing
- Define a security update policy
- Assess the security controls and patch vulnerabilities
- Incident response and recovery
- Improve information sharing amongst industry actors
Liaison & Integration Roles become more Critical and Must be Formalized
Agenda

1. Introduction
2. Current system engineering solutions
3. Future system engineering needs and enhancements
4. Summary and outlook
Summary

- Customer functionality and technical complexity will continue to increase

- Connectivity features with off-board functionality are game changing from different aspects

- Roles and Responsibilities and organizational setup needs to reflect those challenges

- System Engineering methods and tools (e.g. Requirements Engineering) needs to be enhanced to cope with connected features and services

- Cybersecurity needs dedicated design and processes (aligned with Functional Safety and general SW processes)