Scania

Software development and testing

Thomas Gustafsson
thomas.gustafsson@scania.com

Agenda

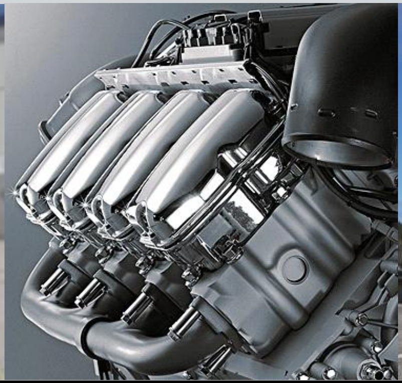
- General info about Scania
- Software development at Scania
- Scania's electrical system
- Integration testing
- Real-time problems and their solutions in industry
    - Some examples from my career

Main message

- Scania is a software company
- Scania has a high degree of in-house development of ECUs
- A lot of freedom and possibilities to learn new things

Corporate statement

Scania's goal is to deliver optimized heavy trucks, buses, engines, and services, offer our customers the best total economy and thereby be the leading company in our business segment. The foundation is Scania's core values, our focus on methods and our motivated coworkers.



Haulage

Construction

Distribution

Special purpose

Network and services

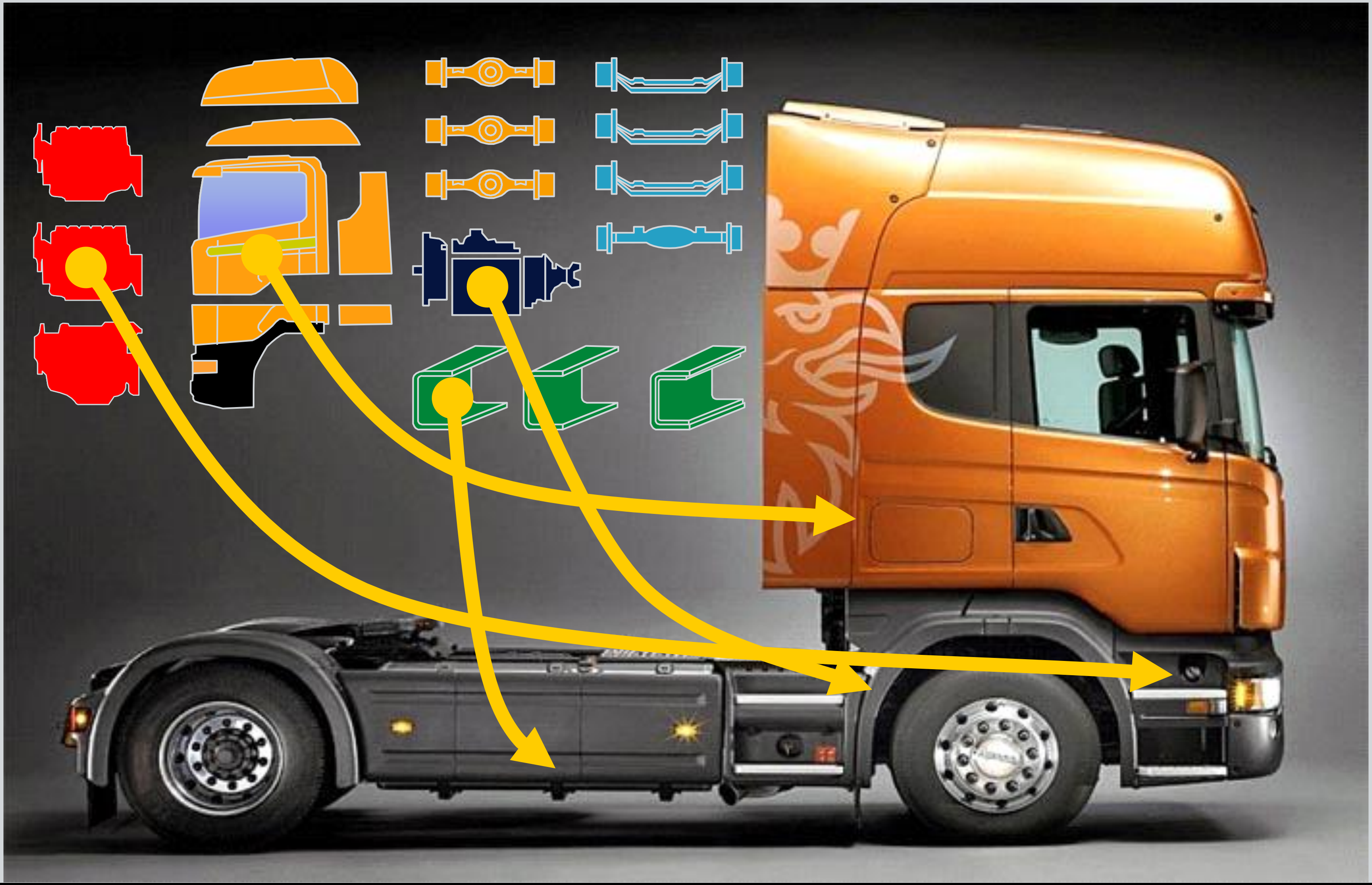
Premium products and services

Intercity and coach

City and intercity

Engines

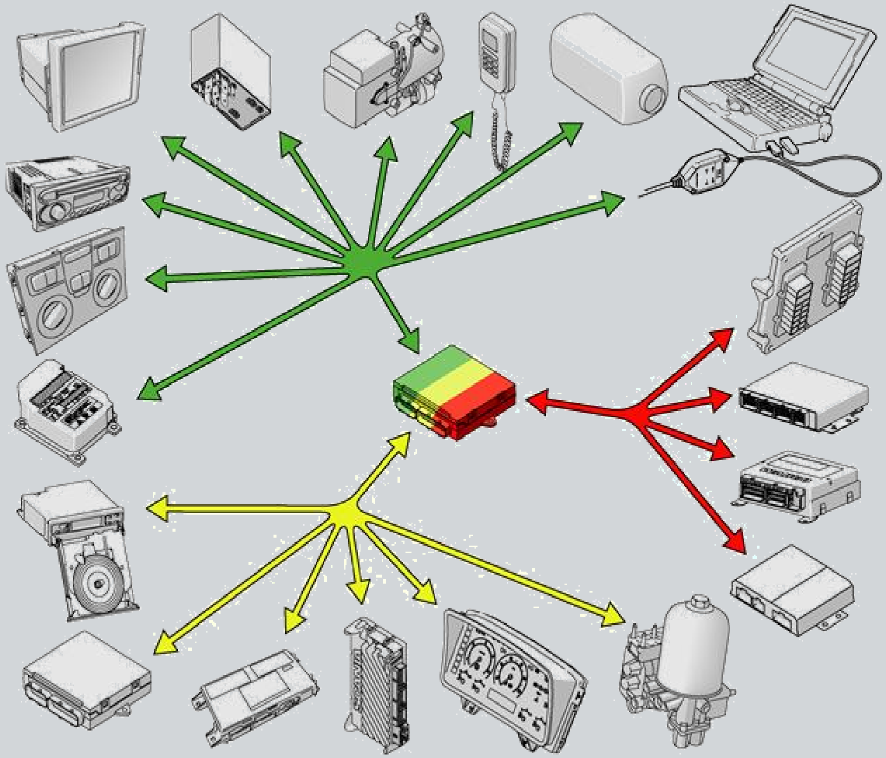
Modular product system

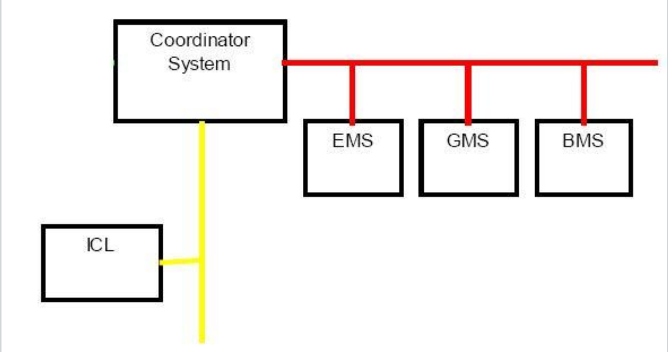


SCANIA

Scania Technical Center

SESAMM – Scanias electrical system





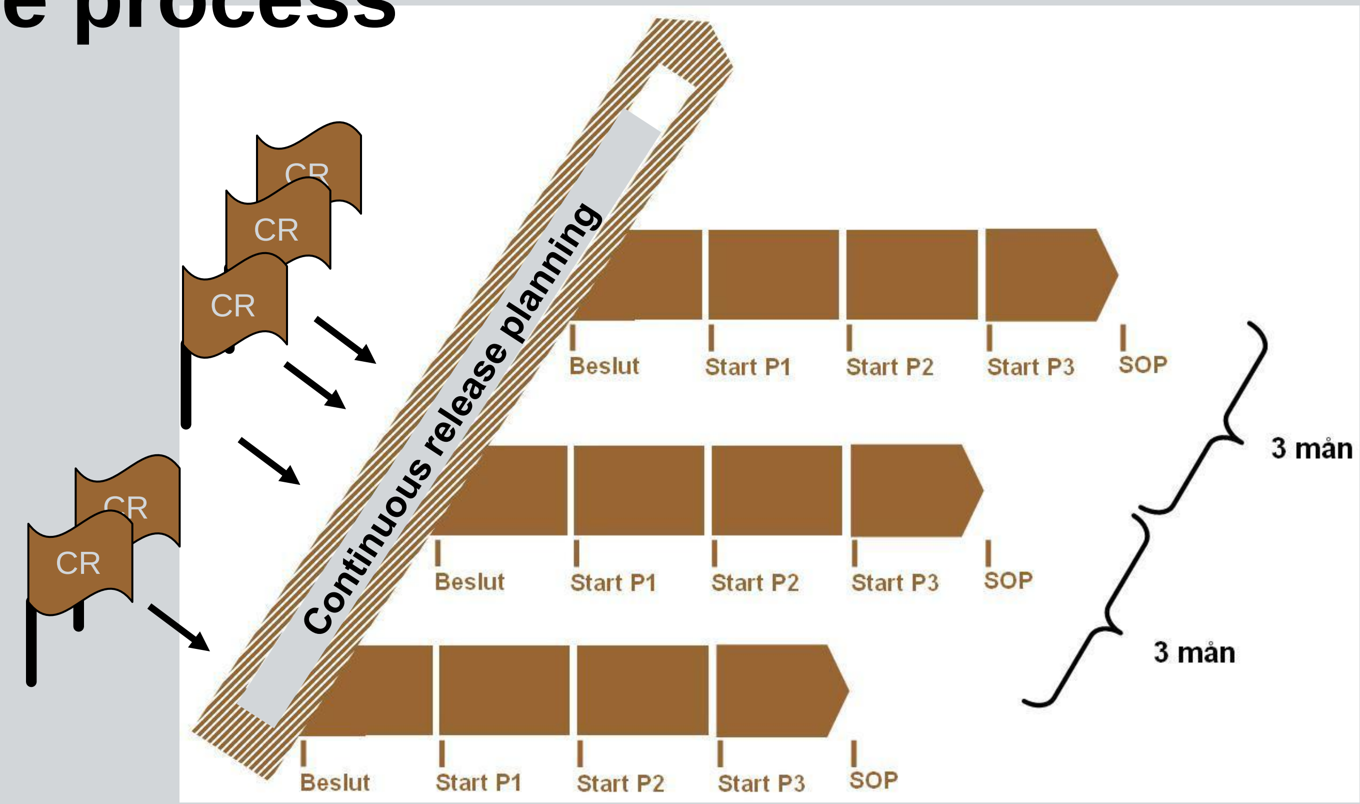
Scania Electrical System - Principles

- One common electrical system for all vehicle types
- Function allocation independent of vehicle specification
- Backward compatible
- Rebuildability
- High level of functionality in degraded mode
- Segments
- In-house development of SW in strategic nodes
- CEPPSS (Continuous Evolution of Properties Planned in Small Steps)

Benefits

- Scalable
  - Few ECUs on low cost vehicles
  - Possibility to add systems and segments for increased content
- Modularised
  - Encapsulation and modularisation reduces communication need and complexity
  - Possible to chose degree of centralisation
  - Clear organisational responsibility for components and functions
- Evolution
  - CEPPSS
  - Balancing complexity and backwards compatibility
- Testing
  - ECU system level testing possible locally before delivery to integration test
  - Stepwise integration possible
- Isolation between ECU systems
  - Easier to prove freedom of interference and avoid unnecessary mixed criticality
- Flexibility in subsegments
  - Possibility to adapt interfaces quickly to new systems without affecting main segments
  - Often in-house SW in main nodes
- Builds on proven concept

Release process



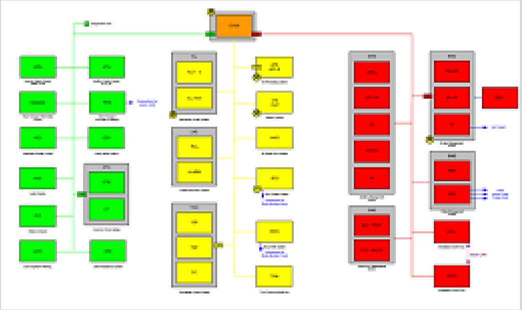
- ✓ Release process is a flow with a pulse
- ✓ Each planned change is flagged with a ***CR***

Abbreviations:
CR = Change Request, P1 = Integr.test 1, P2 = Integr.test 2, P3 =Integr.test 3

Test environments for integration testing

I-lab: Hardware-In-the-Loop

Electrical signals



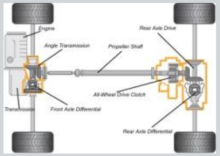
System under test
HW+ SW

Electrical signals

Load simulation | Fault injection

Fault injection

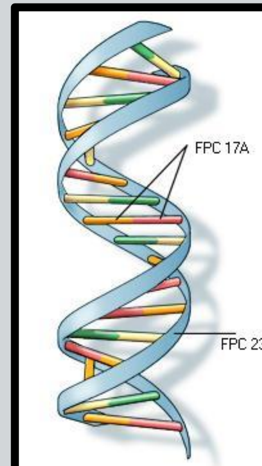
HW



Dynamic vehicle model

Signal handling of sensors

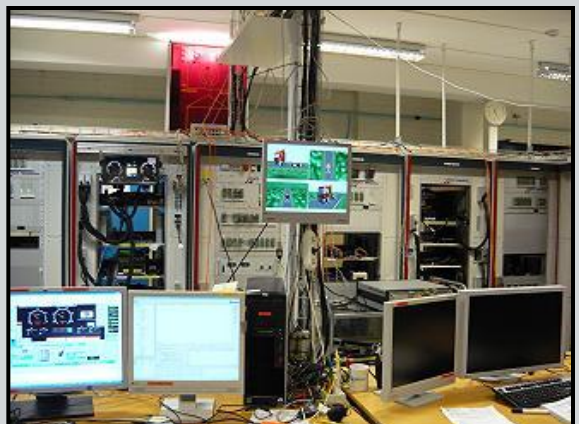
Signal handling of control signals



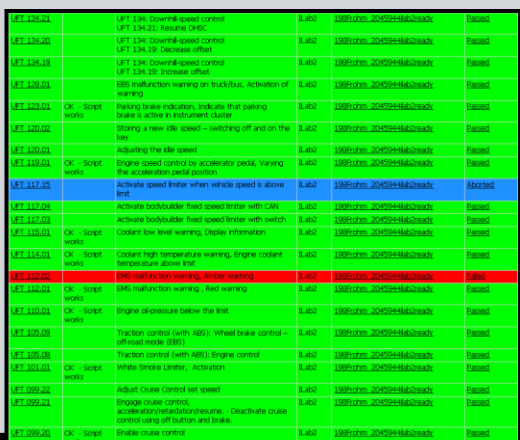
SOPS

+

~180 TC



~250 TC per night



UFT 134.21		UFT 134: Downhill-speed control UFT 134.21: Resume DHSC	ILab2	198Frohm_2045944lab2ready	Passed
UFT 134.20		UFT 134: Downhill-speed control UFT 134.19: Decrease offset	ILab2	198Frohm_2045944lab2ready	Passed
UFT 134.19		UFT 134: Downhill-speed control UFT 134.19: Increase offset	ILab2	198Frohm_2045944lab2ready	Passed
UFT 128.01		EBS malfunction warning on truck/bus, Activation of warning	ILab2	198Frohm_2045944lab2ready	Passed
UFT 123.01	OK - Script works	Parking brake indication, Indicate that parking brake is active in instrument cluster	ILab2	198Frohm_2045944lab2ready	Passed
UFT 120.02		Storing a new idle speed – switching off and on the key	ILab2	198Frohm_2045944lab2ready	Passed
UFT 120.01		Adjusting the idle speed	ILab2	198Frohm_2045944lab2ready	Passed
UFT 119.01	OK - Script works	Engine speed control by accelerator pedal, Varying the acceleration pedal position	ILab2	198Frohm_2045944lab2ready	Passed
UFT 117.15		Activate speed limiter when vehicle speed is above limit	ILab2	198Frohm_2045944lab2ready	Aborted
UFT 117.04		Activate bodybuilder fixed speed limiter with CAN	ILab2	198Frohm_2045944lab2ready	Passed
UFT 117.03		Activate bodybuilder fixed speed limiter with switch	ILab2	198Frohm_2045944lab2ready	Passed
UFT 115.01	OK - Script works	Coolant low level warning, Display information	ILab2	198Frohm_2045944lab2ready	Passed
UFT 114.01	OK - Script works	Coolant high temperature warning, Engine coolant temperature above limit	ILab2	198Frohm_2045944lab2ready	Passed
UFT 112.02		EMS malfunction warning, Amber warning	ILab2	198Frohm_2045944lab2ready	Failed
UFT 112.01	OK - Script works	EMS malfunction warning , Red warning	ILab2	198Frohm_2045944lab2ready	Passed
UFT 110.01	OK - Script works	Engine oil-pressure below the limit	ILab2	198Frohm_2045944lab2ready	Passed
UFT 105.09		Traction control (with ABS): Wheel brake control – off-road mode (EBS)	ILab2	198Frohm_2045944lab2ready	Passed
UFT 105.08		Traction control (with ABS): Engine control	ILab2	198Frohm_2045944lab2ready	Passed
UFT 101.01	OK - Script works	White Smoke Limiter, Activation	ILab2	198Frohm_2045944lab2ready	Passed
UFT 099.22		Adjust Cruise Control set speed	ILab2	198Frohm_2045944lab2ready	Passed
UFT 099.21		Engage cruise control, acceleration/retardation/resume. - Deactivate cruise control using off button and brake.	ILab2	198Frohm_2045944lab2ready	Passed
UFT 099.20	OK - Script works	Enable cruise control	ILab2	198Frohm_2045944lab2ready	Passed

Test report

Real-time in HIL

- The test scripts, running on a PC connected to the HIL, have, normally, no real-time requirements
- All real-time requirements are executed in the processor boards of the HIL

Real-Time in Industry

- Is the theory in the real-time course used in industry?
- How is real-time aspects in software addressed in industry?

- 3 cases
  - Car seat heater
  - Energy meter
  - Bus load in a vehicle
- PINT research project

Car seat heater

- Four step input as an analog value
- Translate into a digital signal and transmit it to an ASIC
- 90% of code is error handling

- The end customer, a big european car manufacturer, had software development process requirements, e.g., the use of MISRA
- But also, how is the event of bursting interrupts handled

Car seat heater

- How do you avoid having to prove that you handle interrupt bursts corrrectly?
  - You don't use interrupts at all!
  - So you use a time-triggered architecture instead, then no need even for timer interrupts
- Slot 1
  - Start free running timer
  - Execute code
  - Wait for timer to reach 4 ms
- …

Car seat heater

- If we can prove that no slot takes longer than 4 ms, then we are safe and have a completely deterministic solution!
- Count assembler instructions and assume for loops take maximum amount of iterations
- We still had a wide margin even though a software implementation of division algorithm

Energy meter

- Accumulate energy transfered in an power grid. Billable quantity which is regulated. The software development process is also regulated

- Strict timeliness requirements
  - Must measure current at fixed sampling intervals relative to start of a sine wave

Energy meter

- Software split into two parts
  - Hard real-time
  - Soft real-time
- Hard real-time is realized using prioritized interrupts
  - Time-triggered approach would be impossible due to the short slot times
  - This part determines current energy and accumulates it
  - Prioritized interrupts is used to increase timeliness
- Soft real-time is realized using tasks in an operating system
  - E.g., HMI and communication protocols

Bus load in a vehicle

- Scania is using results from:
  - R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, “Controller area network (CAN) schedulability analysis: Refuted, revisited and revised,” Real-Time Systems, vol. 35, no. 3, pp. 239-272, 2007.

- This paper highlights that old proofs were wrong, and then corrects them

Bus load in a vehicle

- Response time analysis of CAN frames
- The longest response time shall be found
- It is shown that the *busy period* occurs when the longest lower priority messages has started to be transmitted, at the same time as message *i* and all its higher priority messages are queued for transmission
- Now *i* must wait for the lower priority message and all higher priority messages
- The busy period contains the longest response time
  - Each instance of *i* within busy period must be investigated

Bus load in a vehicle

- The results can also be applied to operating systems scheduling using *cooperative scheduling*
- This approach is attractive in resource constrained embedded systems, since no task specific stack is needed
  - E.g., protothreads and FreeRTOS support cooperative tasks
  - The energy meter has relative many tasks, which shows when totalling used RAM

Master Thesis proposals

- Devise a new programming environment for test script development
  - Today we use Python, but now we have a new test automation framework where we can choose different execution environments for client and server
- Python Offline Debugger
  - Would it be possible to create an offline debugger so the test scripts can be easily debugged without access to the HIL