Voor onze producten komen we niet meer om gebruik van embedded software heen. Hoe voorkom, dan wel beperk, ik dure en tijdrovende ‘leermomenten’ ?”

Gerard Fianen
Indes – Integrated Development Solutions BV

gerard@indes.com www.indes.com 0345-545.535

Mijn achtergrond

Sinds midden 80-er jaren verkoop en ondersteuning ontwikkeltools voor microcontrollers

25 jaar INDES – Integrated Development Solutions BV

- Debug, Test en Verificatie tools voor embedded software
- RTOS, Middelware
- Model Driven Design
- Productie programming
- Advies en lokale ondersteuning

Waarin is Embedded software anders dan ‘gewone’ software ?

Real-Time gedrag, laag stroomverbruik

Beperkt geheugen, OP = OP

Low-level / dicht op de hardware
- Drivers / Flash loader/ verschillende soorten RAM & Flash

On-target debuggen

Safety critical / Certificatie ?

Security & privacy
Kosten van onderhoud / Firmware updates

IoT Security by design

It all starts with the requirements !

ELECTRONICS & APPLICATIONS
14 T/M 16 MEI 2019 JAARBEURS UTRECHT

1) No default passwords ✓
2) Implement a vulnerability disclosure policy
3) Keep software updated ✓
4) Securely store credentials and security-sensitive data ✓
5) Communicate securely ✓
6) Minimise exposed attack surfaces ✓
7) Ensure software integrity ✓
8) Ensure that personal data is protected ✓
9) Make systems resilient to outages ✓
10)Monitor system telemetry data
11)Make it easy for consumers to delete personal data ✓
12)Make installation and maintenance of devices easy ✓
13)Validate input data ✓
14) Keep verifying for vulnerabilities

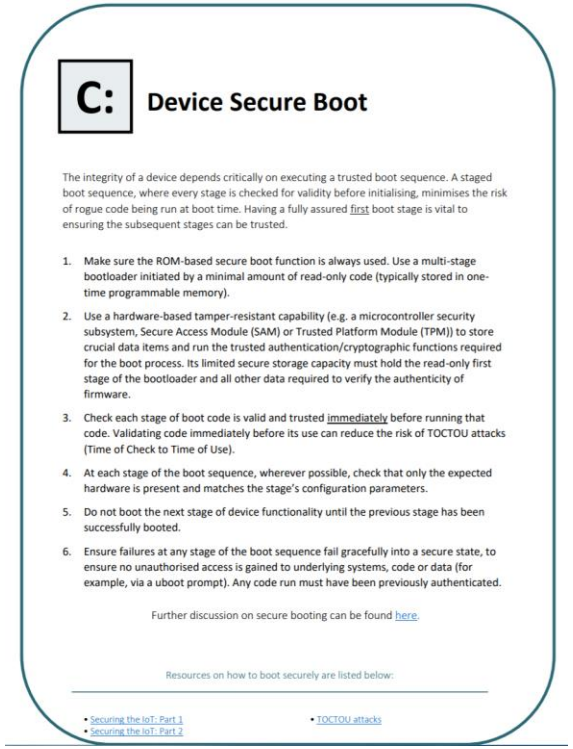
The right tools simplify Secure by Design

Seminar / workshop : www.embedded-masterclass.nl

Best Practices guides

IoT Security Foundation

IoT Security Foundation www.iotsecurityfoundation.org
– Initially UK-centric, now the global forum for IoT Security



C: **Device Secure Boot**

The integrity of a device depends critically on executing a trusted boot sequence. A staged boot sequence, where every stage is checked for validity before initialising, minimises the risk of rogue code being run at boot time. Having a fully assured first boot stage is vital to ensuring the subsequent stages can be trusted.

1. Make sure the ROM-based secure boot function is always used. Use a multi-stage bootloader initiated by a minimal amount of read-only code (typically stored in one-time programmable memory).
2. Use a hardware-based tamper-resistant capability (e.g. a microcontroller security subsystem, Secure Access Module (SAM) or Trusted Platform Module (TPM)) to store crucial data items and run the trusted authentication/cryptographic functions required for the boot process. Its limited secure storage capacity must hold the read-only first stage of the bootloader and all other data required to verify the authenticity of firmware.
3. Check each stage of boot code is valid and trusted immediately before running that code. Validating code immediately before its use can reduce the risk of TOCTOU attacks (Time of Check to Time of Use).
4. At each stage of the boot sequence, wherever possible, check that only the expected hardware is present and matches the stage's configuration parameters.
5. Do not boot the next stage of device functionality until the previous stage has been successfully booted.
6. Ensure failures at any stage of the boot sequence fail gracefully into a secure state, to ensure no unauthorised access is gained to underlying systems, code or data (for example, via a uboot prompt). Any code run must have been previously authenticated.

Further discussion on secure booting can be found here.

Resources on how to boot securely are listed below:

- Securing the IoT: Part 1
- Securing the IoT: Part 2
- TOCTOU attacks

ARM BT BarcoSilex dyson NXP Imagination chipless SECURE THINGZ
infineon Visteon vodafone intel Driving Trust inside SECURE THALES SAMSUNG WEBROOT

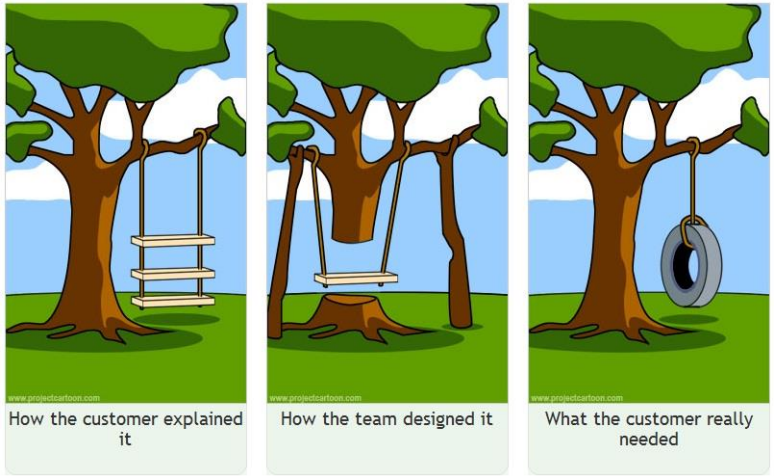
Over 100 members and growing...

Uitbesteden?

Requirements, requirements, requirements (communicatie)

Besteed niet de core-kennis / differentiator van je product uit
- Houd deze kennis binnenshuis

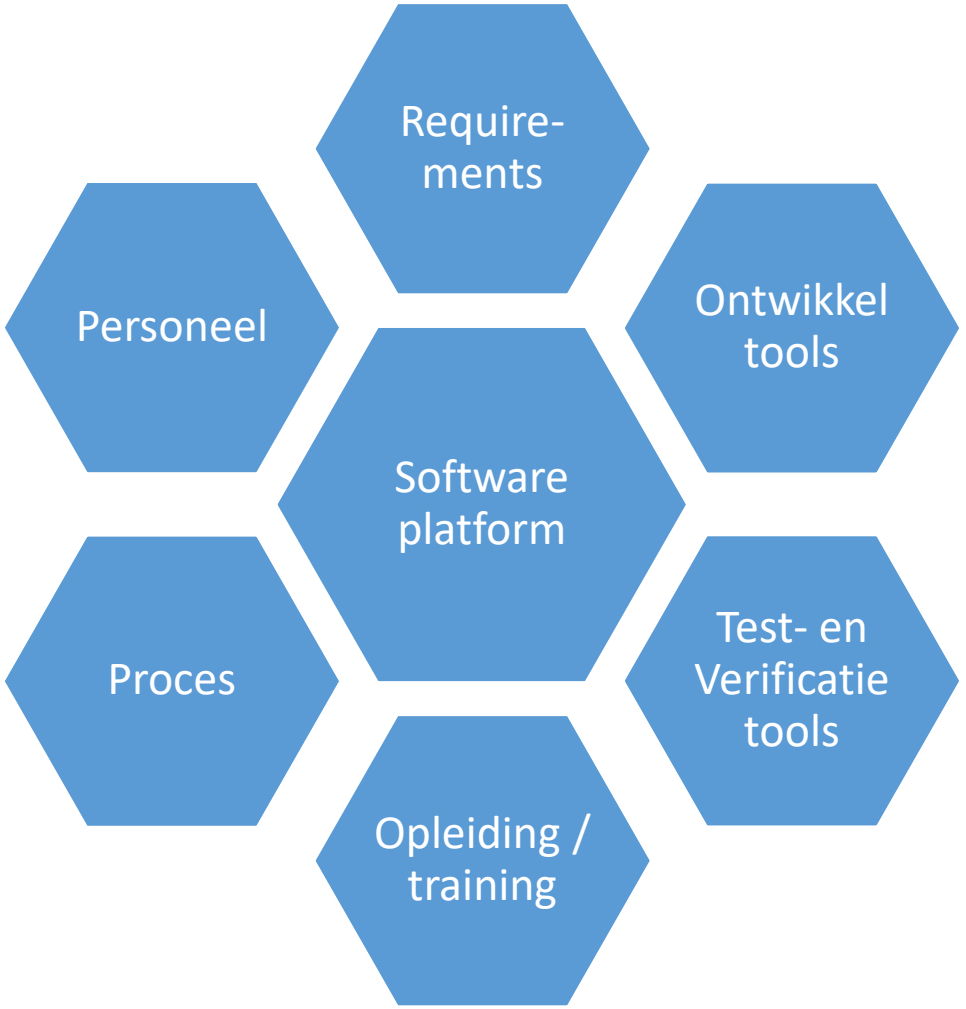
Zorg dat je eigenaar bent van door een partner gemaakte code



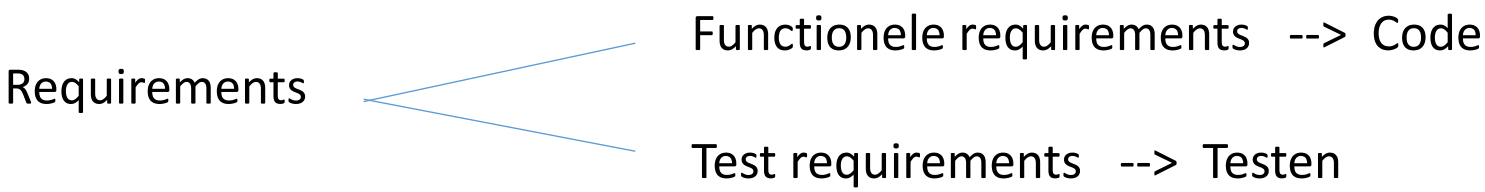
Reproduceerbaarheid :
- Source- en binaire code (natuurlijk ook van de hardware zoals CAD & FPGA files).
- Toegang tot de gebruikte ontwikkel- en test tools
- Documentatie
- Archiveren

Realistisch budget voor de software

Software In-House ontwikkelen



Requirements, Test & Verificatie --> Kwaliteit



Safety critical : Requirements Traceability Matrix (RTM)

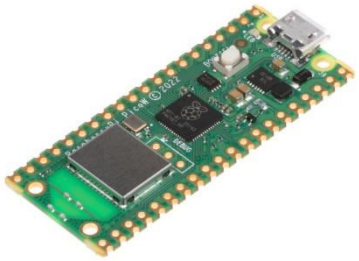
Hardware

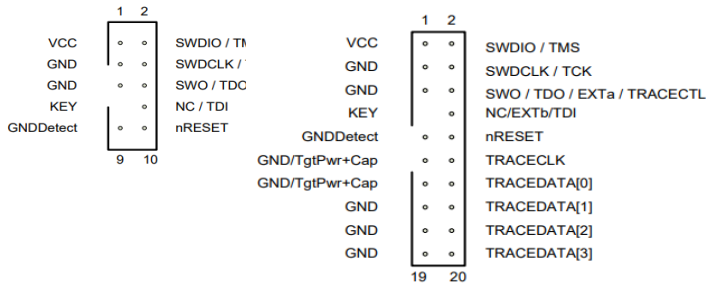
Welke Microcontroller familie ?
- Denk ook aan beschikbaarheid en EOL garanties.

Deel in een FPGA ?

Eigen hardware of een standaard MCU/SOM module
(SBC, Raspberry Pi, Arduino)

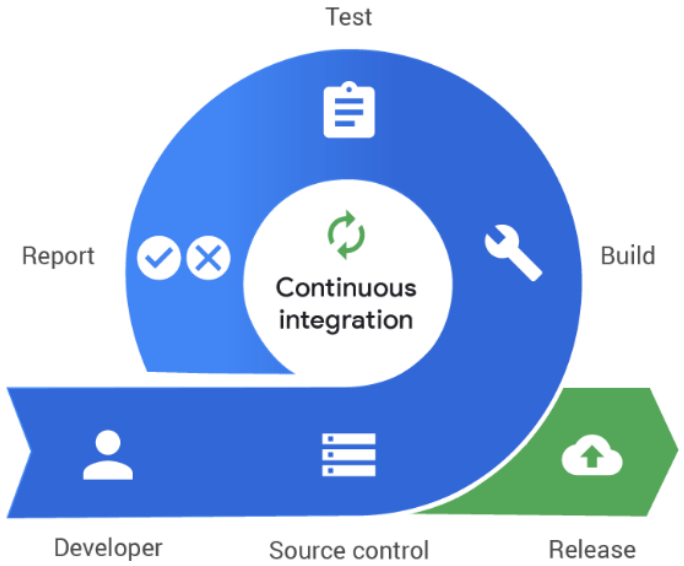
Vergeet de debug & programmeer interface niet !
- J-TAG/SWD of ETM/Nexus





Personeel en proces

Embedded software vraagt meer van S/W engineers dan ‘gewone’ software
Programmeer taal : C / C++ / Embedded C++ (Python, Rust)
Model Driven Design , Automatische code generatie ?



Begeleid de S/W engineers in het volgen van een proces
Kwaliteit, onderhoudbaar en reproduceerbaar

- Begin vanuit de S/W requirements
- Niet te slimme / cryptische code maar leesbare code
- Comments en documentatie
- Programming Standards Verificatie (b.v. MISRA)
- Testen (unit Test / Functionele test)

Blijf steeds bewust dat S/W engineers op enig moment het bedrijf verlaten
- Documentatie, documentatie etc

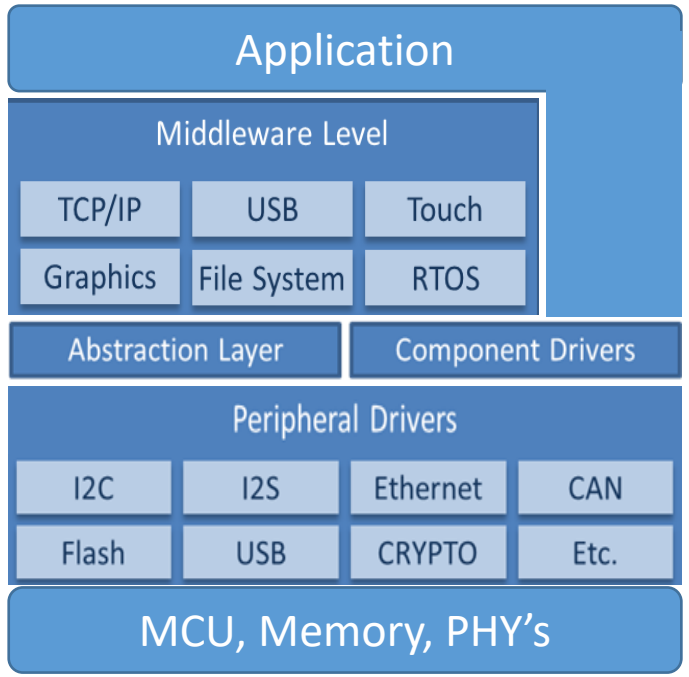
Software Platform (RTOS)

Selecteer een Software Platform dat past bij applicatie & engineers
- Multitasking nodig ?
- ‘Gratis’ of commercieel ?
- Linux ?

Welke standaard modulen zijn nodig ?
- Denk aan Networking, Filesysteem, Security, GUI
- Eigendomsrecht code, pas op voor ‘slapende’ IPR
- Expert support
- Drivers

Security:
Secure protocollen, privacy wetgeving

Firmware updates



Operating System

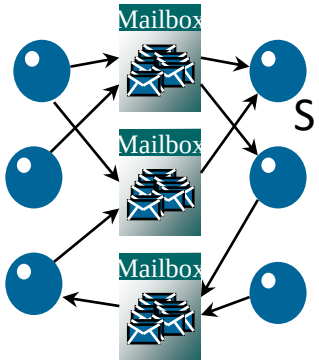
Through-put optimized

Linux
Windows
Android
iOS

Let op :
- External memory / FS
- Niet hard Real-Time
- Boot time

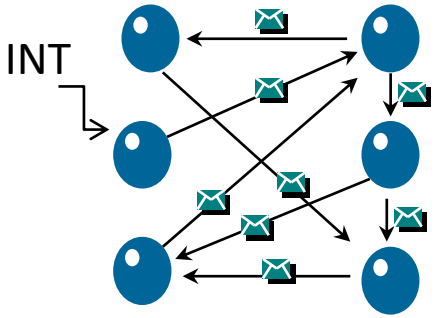
Response Time optimized (RTOS)

Classic / mailbox



FreeRTOS
SEGGER embOS
VxWorx
ThreadX
GreenHills

Direct Message passing



QNX
Sciopta
OSE

Home Brew

Embedded Development Tools

Evalueer de tools waar mogelijk in de embedded omgeving

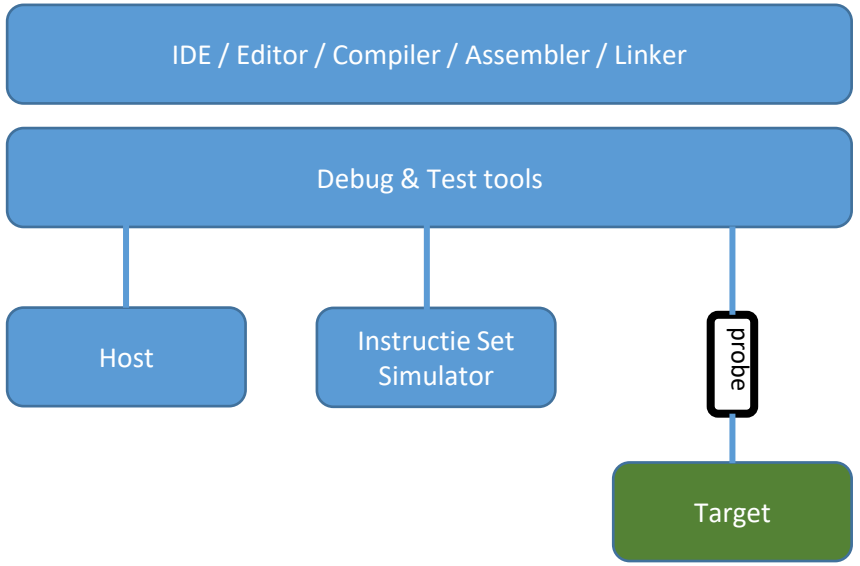
Pas op voor vendor / MCU lock-in (porteerbaarheid)

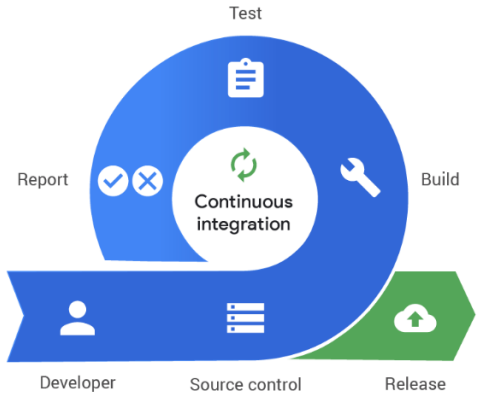
Geïntegreerde Programming Standards Verificatie
Geïntegreerde Test en verificatie

Reproduceerbaarheid

- Versie controle (CI/CD)
- Regressie testing
- Archivering, toegang tot oude versies

Support van leverancier bij installatie en integratie





Test & Verificatie - Kwaliteit

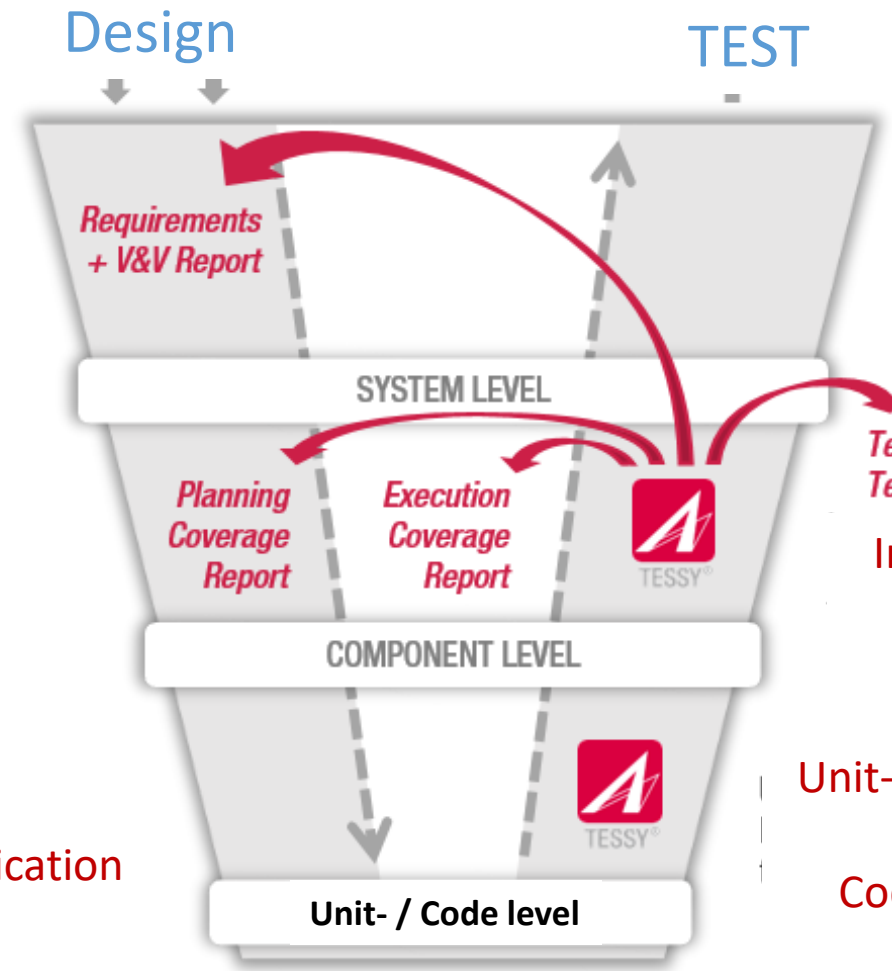
System level

Requirements

Module / subsystem level
Module requirements

Unit - code level

Programming Standard Verification
Static Analysis



System Test

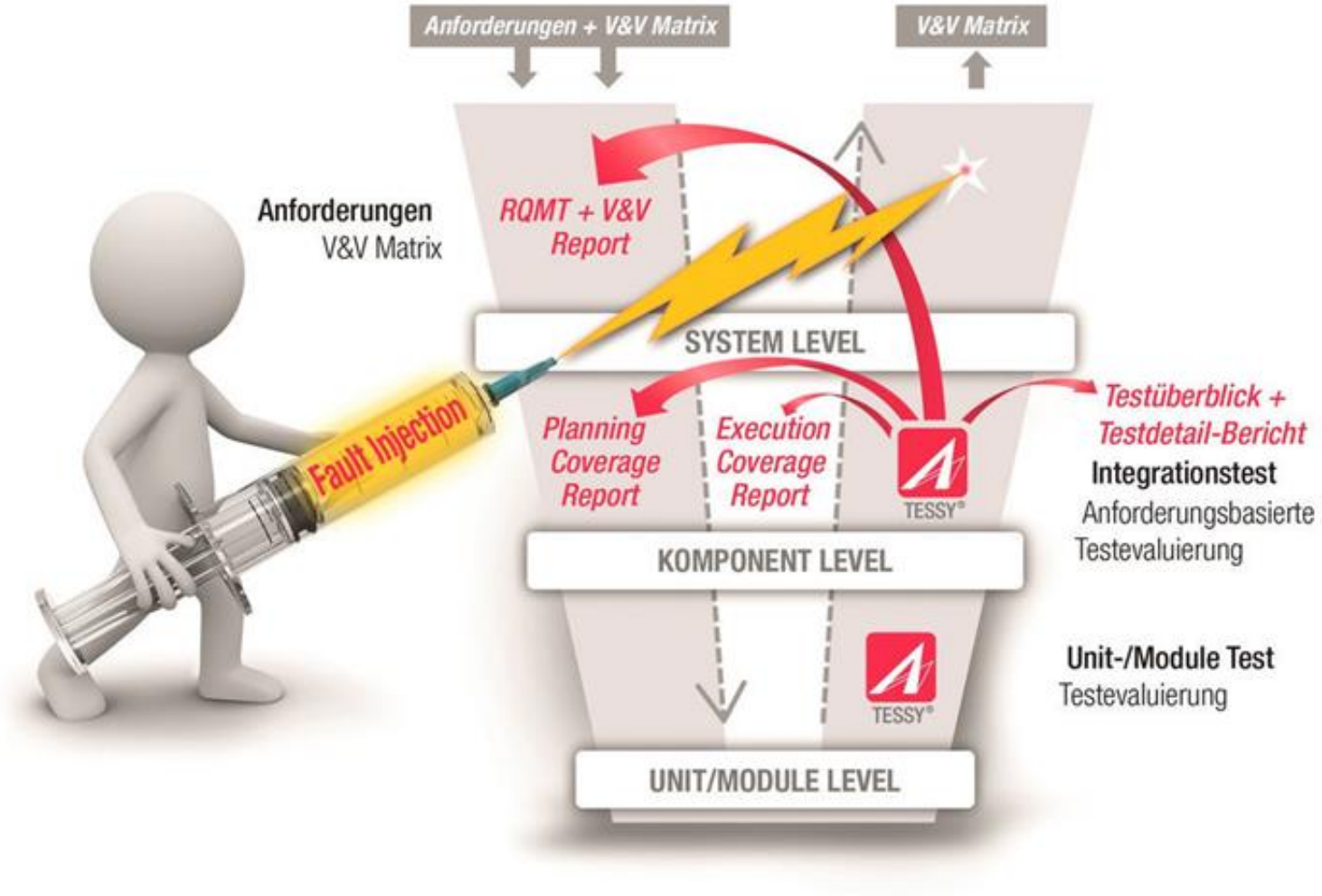
Test Overview Report
Test Details Report

Integratie Test

Unit- / module Test

Code Coverage

Test & Verificatie - Kwaliteit



INDES –
Integrated Development Solutions BV

Cross Compilers, Debuggers, IDE
RTOS, Middleware, Protocol stacks, Security, GUI, EFS
Debug & Trace probes, Emulators
Real-Time Trace, RTOS-Event Trace
Static Analysis, Timing Analysis, Stack Analysis
Unit Test, Code Coverage
Flash Programming

On-site support
Test-as-a-Service
Verification-as-a-Service

www.indes.com info@indes.com Tel: 0345 - 545.535