



E-PLM 2.0



Experiment 3.3:
Hardware – software koppeling
Eindpresentatie

www.smartindustry.nl

Producten zijn vaak samengesteld met zowel hardware als software. Deze worden door verschillende afdelingen/leveranciers geleverd. In het product komt dit bij elkaar. In grote systemen worden zelf pas bij inschakelen, de subsystemen voorzien van de juiste software door de distributie vanuit het overkoepelende systeem. Daarnaast kan tijdens de nazorg zowel de hardware als software vervangen worden door nieuwere producten, die door technologische veranderingen aanpassingen hebben. Denk aan vernieuwde processors en upgrades van software.

Problematiek is dat met de huidige configuratie management methodieken de relatie van de oorspronkelijke hardware en software configuratie methodieken geen eenduidige relatie tussen hard en software wordt gelegd.

Centrale vragen:

Welke HW-SW typeringen zijn er te maken die een verschillende aanpak nodig hebben?

Zijn er standaarden en oplossingen voor gecombineerde configuratie van hardware en software?

Wat is nodig om de koppeling op langere termijn zeker te stellen?

De bovengenoemde problematiek geldt zowel voor grote als kleine systemen. Het is daarom voor technologische productontwikkelaars, DEMCON als Thales belangrijk deze problematiek te beheersen. Gezamenlijk willen we de volgende doelen bereiken:

Eenduidige relatie tussen hardware en software vastleggen bij configuratie identificatie.

Borgen dat tijdens change management de relatie tussen hardware en software geborgd blijft.

Thales en DEMCON onderzoeken en overleggen welke methodieken er zijn om de doelstellingen te bereiken. Binnen dit E-PLM 2.0 experiment worden uiteindelijk de doelstellingen getoetst in de praktijk. Zowel Thales als DEMCON kiezen een product uit hun portfolio om dit te gebruiken als casus.

Het experiment

Doel van het experiment

Het doel hardware – software koppeling experiment is opgesplitst in volgende subdoelstellingen en bijbehorende deelexperimenten.

Doelstelling 1 – Overzicht van HW-SW typeringen en mogelijke oplossingen voor gecombineerde configuratie van hardware en software

Een vooronderzoek naar alle mogelijkheden van gecombineerde configuratie van hardware en software afgezet naar de verschillende hardware – software typeringen die er zijn.

Doelstelling 2 – Eenduidige relatie tussen hardware en software vastleggen

De relatie tussen hardware en software op een eenduidige wijze vastleggen. Dit niet alleen tijdens het ontstaan van het product (configuratie identificatie) maar ook tijdens zijn levensduur (change management). Naast het kiezen van de juiste methodiek, procedures en systemen wordt dit ook in de praktijk getoetst.

Werkwijze en methodiek

Bij de start van het experiment werd een kick-off meeting georganiseerd met alle deelnemende partijen. Er vond tweemaandelijks een fysiek overleg tussen plaats tussen de partners (bij DEMCON Enschede of Thales).

Het verzamelen en archiveren van projectinformatie gebeurt op de e-TOL portal van Thales.

DEMCON is de trekker van het experiment en is verantwoordelijk voor het behalen van de milestones. De overige partners leveren voornamelijk informatie aan en reviewen de resultaten

DEMCON verzorgt de eindrapportage.

In het kader van het project wordt mogelijk vertrouwelijke informatie uitgewisseld. De opgestelde rapportages geven enkel een analyseresultaten weer en bevatten geen vertrouwelijke inhoud.



Resultaten

- Bij eerste traject hebben we een overzicht opgesteld welke configuratie methodiek bij welk soort software-hardware combinatie past.
- Bij het tweede traject hebben we ons geconcentreerd op COTS (commercial of the Shelf) producten en welke praktijk problemen we hierbij ondervinden.

Deliverables

Doelstelling 1 (overzicht hardware-software configuraties methodieken)

Doelstelling 2 (Welke configuratie methodiek te kiezen bij COTS producten)

Software typing	Configuration methodology /Versie control	Comments
Commercial off the shelf (COTS)	Buy item (no version control)	Is this always the best approach?
	Composed part (part revision)	If we run by the PCB supplier it is more convenient to have an item for the firmware and one for the PCB. The parent mother than the combination of the two that should be ordered.
embedded software	Software release (stand-alone item)	If we do the flashing in our own production it is more convenient to treat this as a separate software release. Is at FPGA so done until now.
Operating Systems (OS)	Software release (stand-alone item)	item of the software components in the baseline
Functional software	Software release (stand-alone item)	item of the software components in the baseline
System dependent software	Software release (stand-alone item)	Is not at the baseline but kept with the as maintained version
		Is actually no link hardware but software-software link. Functional software vs. OS. You want up here a virtual environment for or at least a document describes the building environment
Building software	Software release (stand-alone item)	
Test & production software	Software release (stand-alone item)	item of the software components in the baseline, where it is not a delivery of the product itself



- Gebruik ook hier part revisie.
- Maak goede afspraken met de leverancier over communicatie van wijzigingen

Lessons Learned & best practices

Voor de koppeling van hardware en software in producten hebben we in de context de volgende vragen centraal gezet:

Welke HW-SW typeringen zijn er te maken die een verschillende aanpak nodig hebben?

Zijn er standaarden en oplossingen voor gecombineerde configuratie van hardware en software?

Wat is nodig om de koppeling op langere termijn zeker te stellen?

In eerste instantie (doelstelling 1) hebben we een overzicht gemaakt van alle mogelijke combinaties tussen hardware en software onderdelen. In het algemeen heeft het de voorkeur zoveel mogelijk hardware en software hun eigen leven te laten leiden. (hardware onderdeel van de BOM. Software onderdeel van software release, die pas op systeemniveau bij elkaar komen.)

Uitzondering zijn hierop de COTS (Commercial of the shelf) producten. Deze worden door leveranciers als 1 artikel geleverd. Uit een praktisch experiment (doelstelling 2) is geconcludeerd dat niet een juist versiebeheer/configuratie methodiek de doorslag geeft, maar duidelijke afspraken over de communicatie van wijzigingen in het product.

Volgende aanbeveling bij COTS producten (zeker als er een seriële software interface is met je eigen te ontwikkelen software):

Stel bij engineering al een specificatie op hoe om te gaan met wijzigingen/nieuwe software releases en hoe deze te communiceren. (Purchase Specification Document)

Laat inkoop deze specificatie meeleveren bij inkoop van het onderdeel en verder afstemmen met de leverancier.