DESIGNING A SMART AUTOMOTIVE FRONT LIGHTING SYSTEM FOLLOWING THE ASPICE PROCESS

Pulleti Sohil Manikanta
Senior SE-Advocate, Blue-Kei Solutions Pvt. Ltd, INDIA.
E-mail: sohilpulleti@blue-kei.com

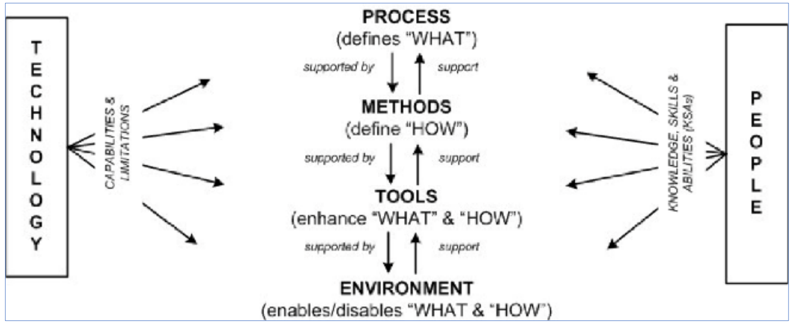
Digital Engineering with Blue-Kei Solutions

- **To** empower decision makers in transforming businesses digitally
- **By** helping to manage cost & schedule and efficiently develop systems,
- **Using** scientific SE methods and systemic approaches
- **While** maintaining design integrity and minimal rework.

15+
Fortune 500 Companies Served

25+
Industry Experts

85%
Repeat Customers



Aerospace	Automotive	Satellite	Manufacturing	MedTech	Railways
MBSE for Route to Certification	**MBPLE**	**Requirements Engineering**	**SE for Innovation**	**Variant Management**	**MBSE for Functional and Safety**

-- Offerings --

Knowledge Consulting
Compliance to ISO15288, ASPICE, ISO26262,
Reporting through stage-gate process

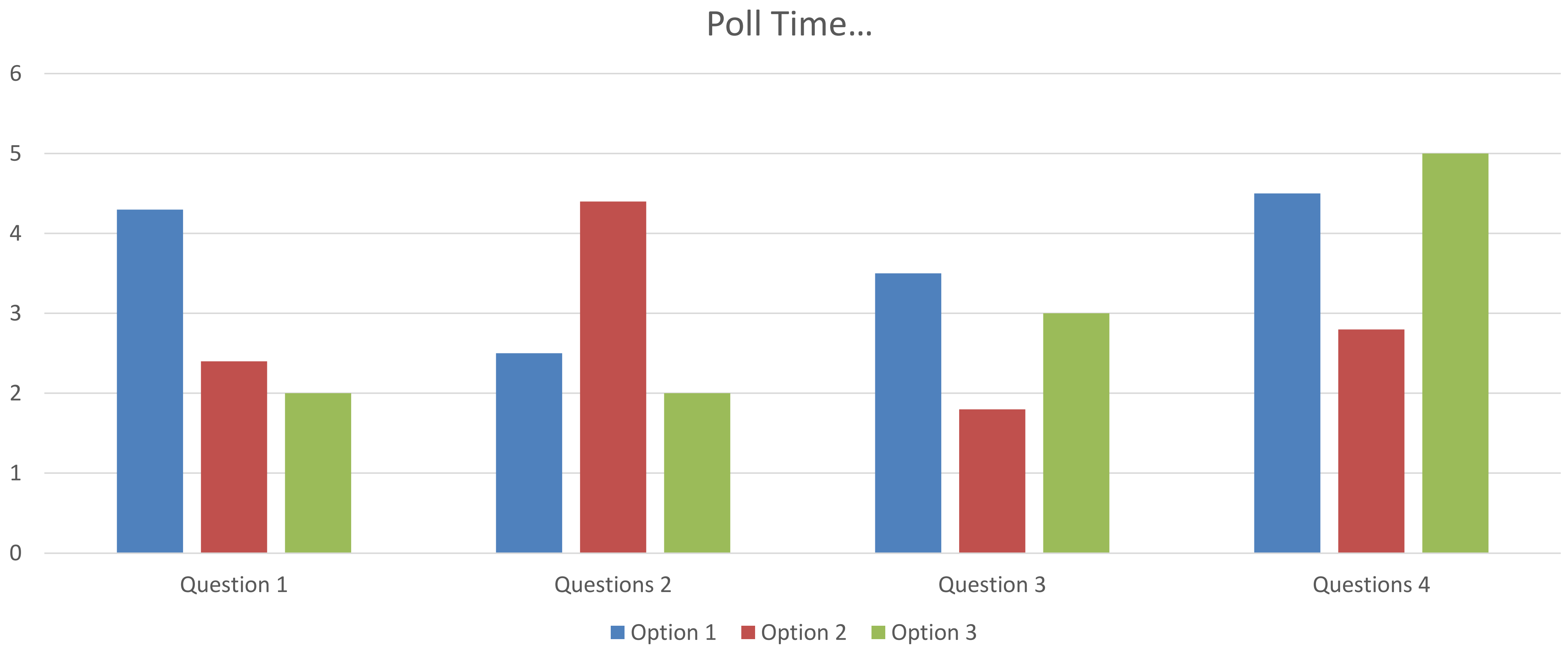
Competency Development
Experiential workshops on fundamentals & Masterclasses

Project Execution
SE, MBSE process & method implementation

AGENDA

- Introduction
- Challenges
- Approach: ASPICE process using Capella
- Value Additions & Benefits
- Conclusion

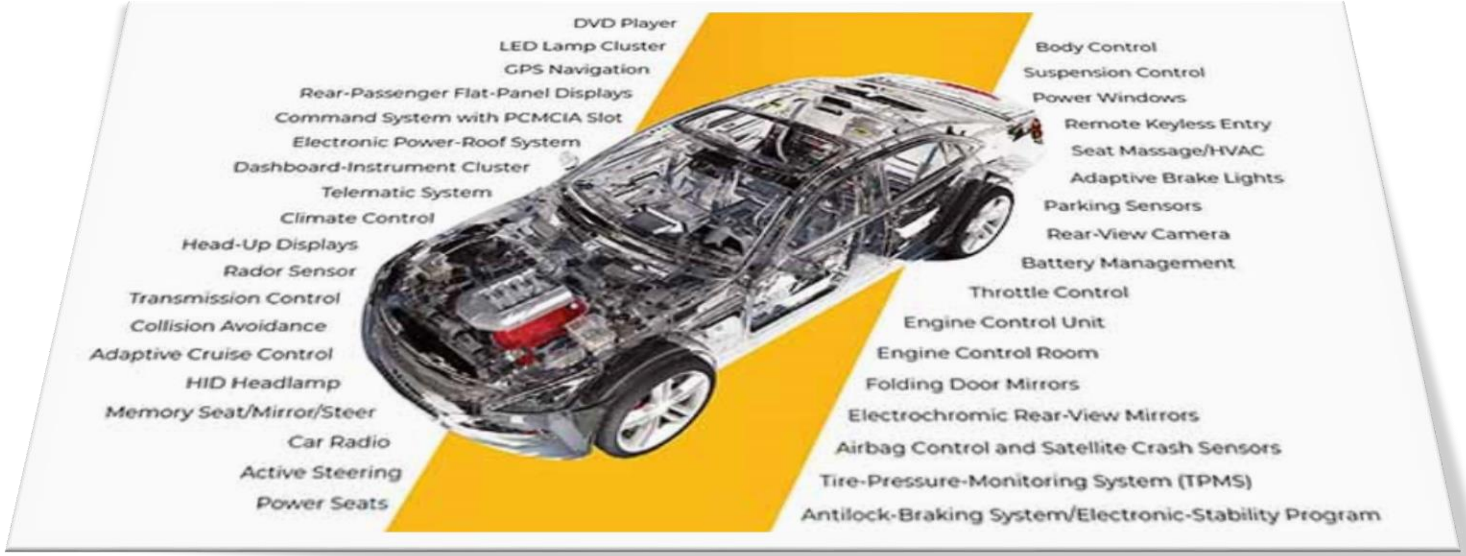
Let's understand the Room:



Product Development Challenges in Automotive Systems

- Key challenges:
  - Lack of end-to-end traceability
  - Hidden assumptions between hardware, software, and sensors
  - Lack of system-level validation before implementation
  - Difficult reuse of legacy systems
  - Supplier misalignment & integration issues

Source: AI Generated

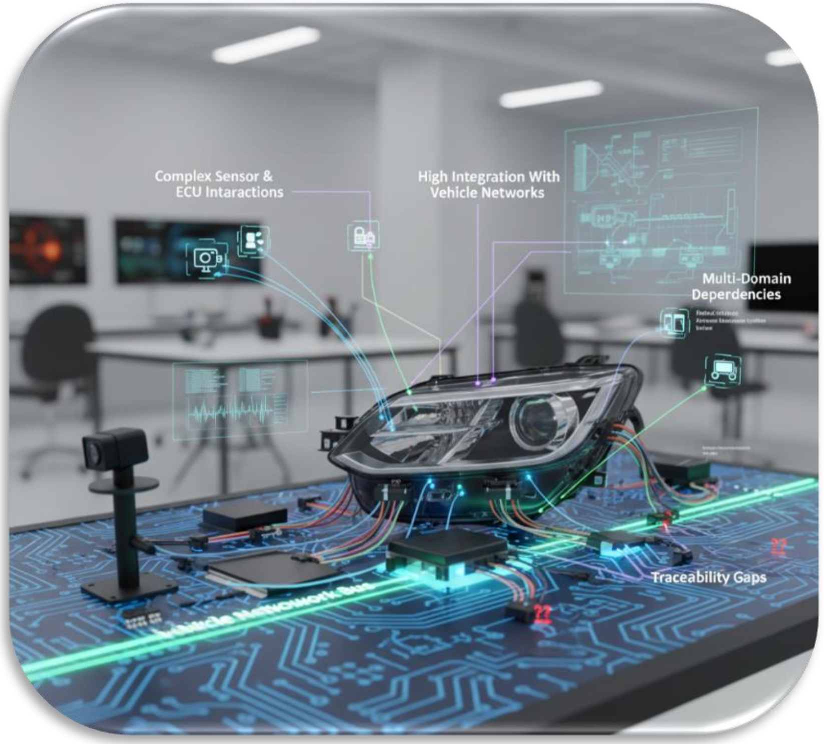


Source: https://www.ansys.com/blog/4-ways-to-design-reliable-automotive-electronics

Why Automotive Lighting System?

Includes mechanical, electrical, control, and software domains and has strict regulatory and safety constraints

- A perfect example of MBSE complexity.
- Using a Smart Automotive Lighting System case study, we demonstrate how a model-based approach supports ASPICE processes by improving clarity, traceability, and engineering quality
- Common issues:



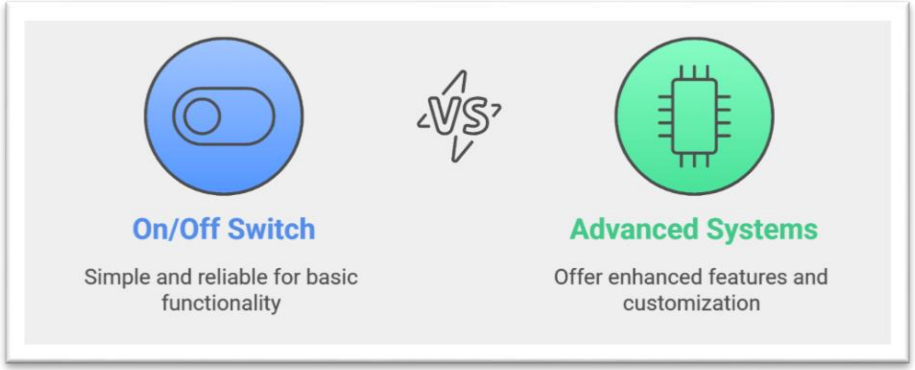
Source: AI Generated

Source: AI Generated

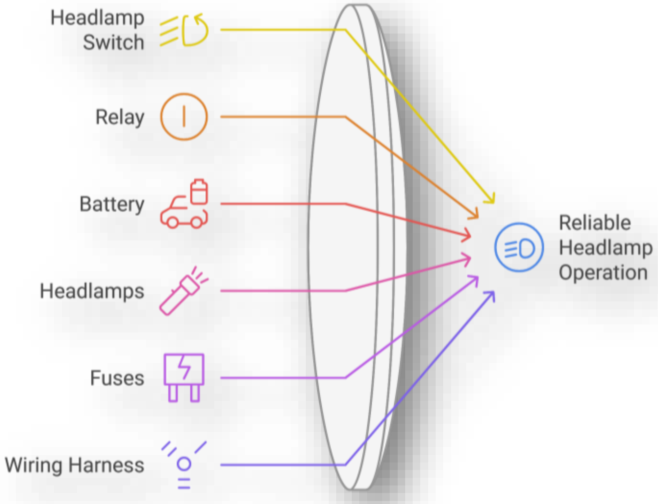
Automotive Lighting Systems

ON/OFF Switch Lighting System

Adaptive Lighting System

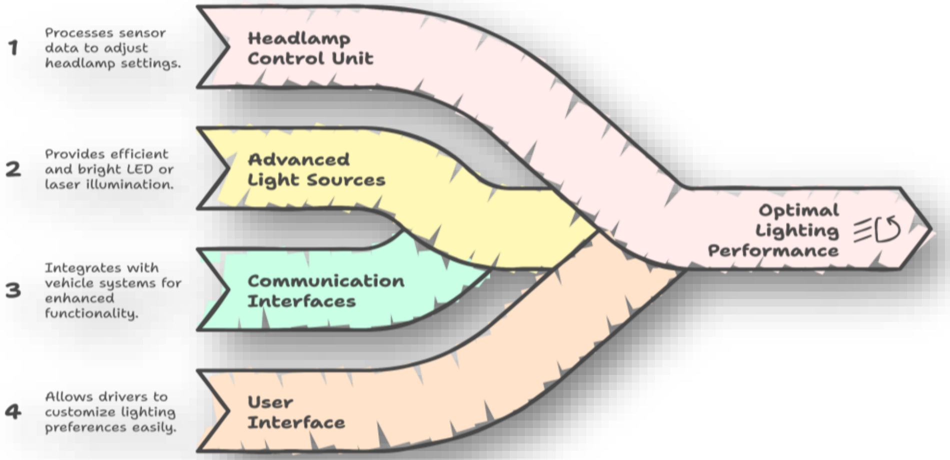


Source: AI Generated



Source: AI Generated

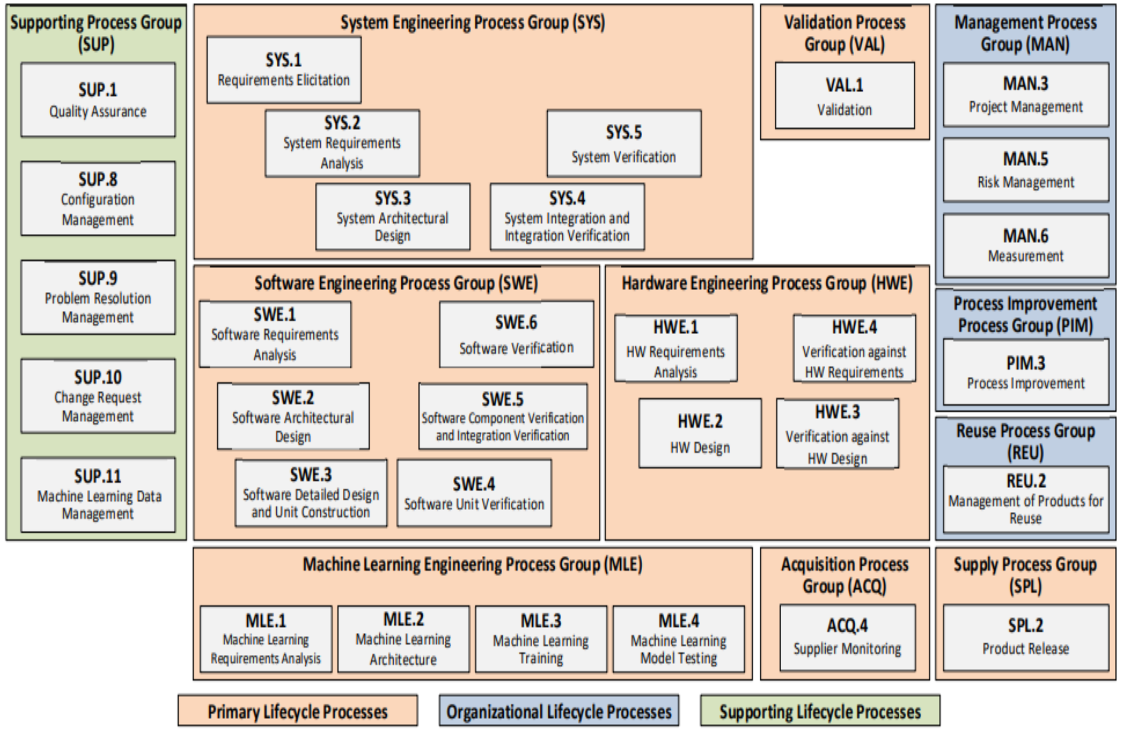
Transition from Simple ON/OFF Switch to Adaptive Lighting Systems



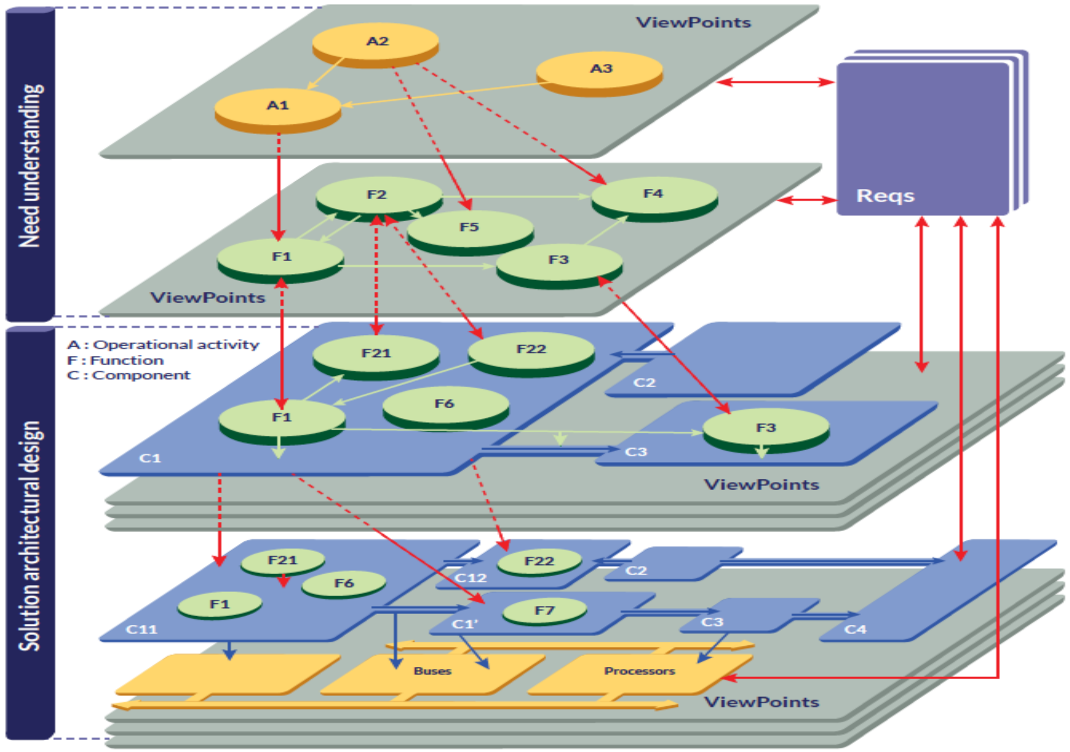
Source: AI Generated

ASPICE meets ARCADIA

- ❖ This presentation shows how Arcadia/Capella's OA → SA → LA workflow matches the ASPICE SYS.1–SYS.3.
- ❖ ASPICE defines the required engineering processes to ensure maturity, consistency, and traceability.
- ❖ Modelling all system attributes within this workflow results in clear requirements, well-defined architectures, and strong traceability improving both engineering quality and ASPICE readiness.



Source: https://vda-qmc.de/wp-content/uploads/2023/12/Automotive-SPICE-PAM-v40.pdf



Source: https://mbse-capella.org/

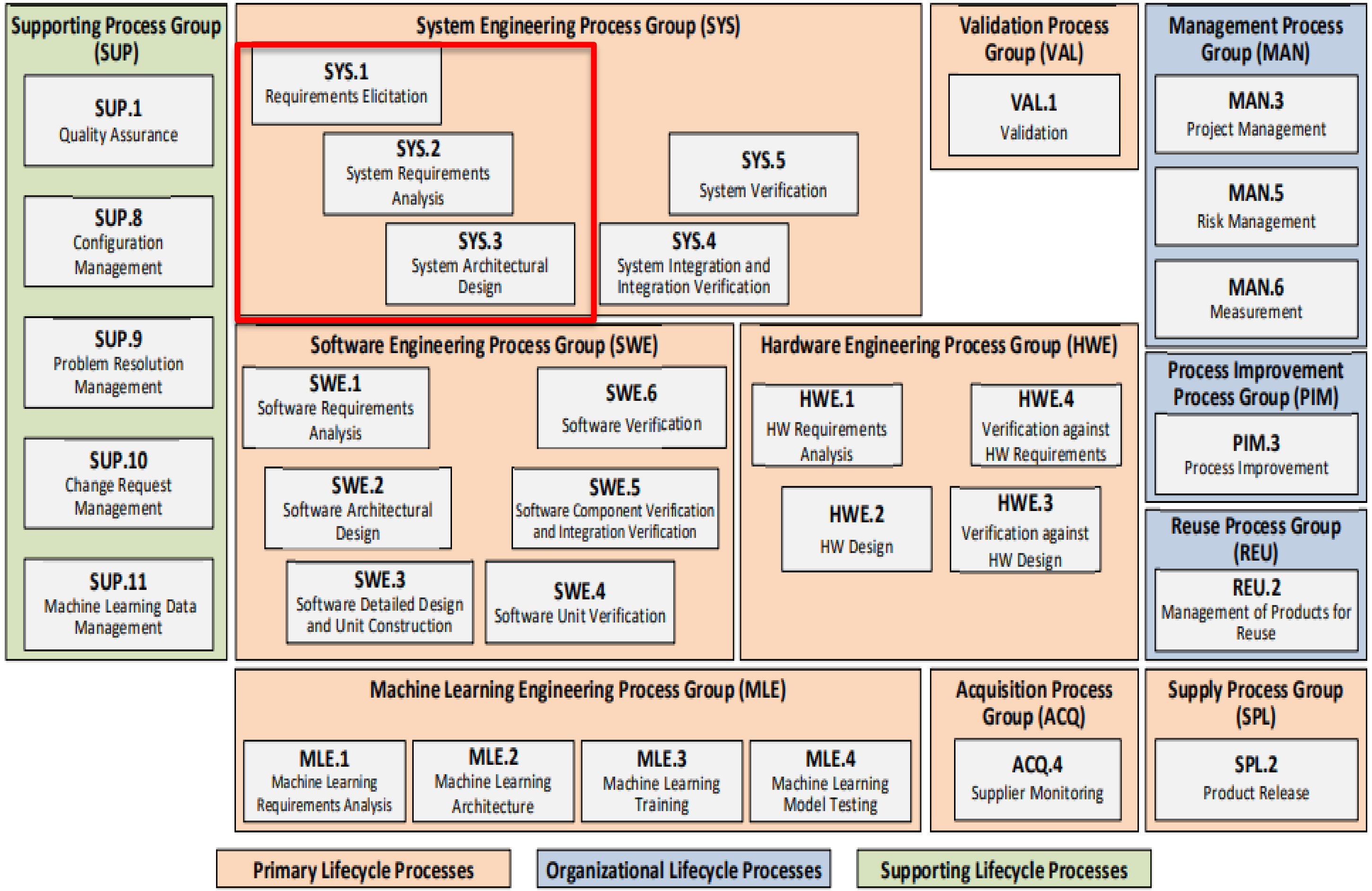
What is ASPICE?

- ASPICE is a Process Assessment Model(PAM) and a Process Reference Model(PRM) for system, software, hardware, mechanical, and cybersecurity development in the automotive industry.

- Teams who design and develop products for the automotive industry increasingly use the ASPICE model to improve and optimize development processes and measure organizational process maturity.

Challenges in ASPICE:

Hidden Risks of ASPICE Non-Compliance
Late integration issue
Failed process audit
Traceability gap
Poor requirements management

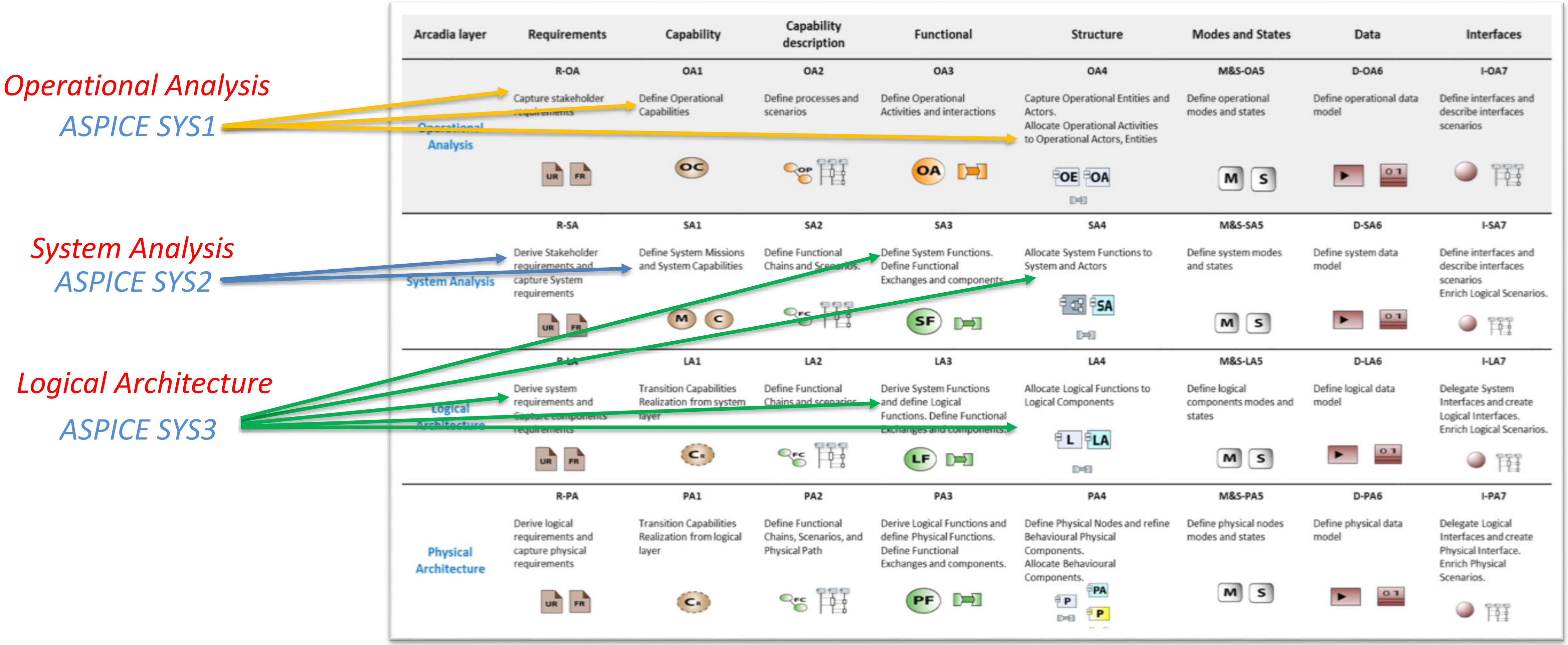


Source: https://vda-qmc.de/wp-content/uploads/2023/12/Automotive-SPICE-PAM-v40.pdf

The ASPICE Process with Arcadia

BLUEKEI SOLUTIONS

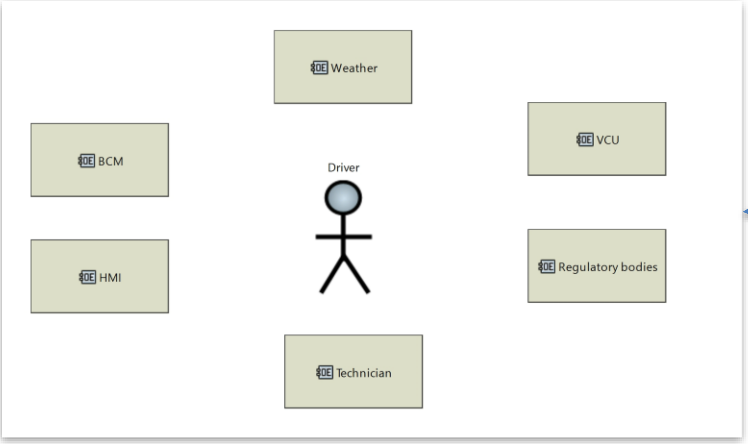
❖ Arcadia methodology – a structured, model-based engineering framework:



Arcadia layer	Requirements	Capability	Capability description	Functional	Structure	Modes and States	Data	Interfaces
Operational Analysis	R-OA Capture stakeholder requirements	OA1 Define Operational Capabilities	OA2 Define processes and scenarios	OA3 Define Operational Activities and interactions	OA4 Capture Operational Entities and Actors. Allocate Operational Activities to Operational Actors, Entities	M&S-OA5 Define operational modes and states	D-OA6 Define operational data model	I-OA7 Define interfaces and describe interfaces scenarios
System Analysis	R-SA Derive Stakeholder requirements and capture System requirements	SA1 Define System Missions and System Capabilities	SA2 Define Functional Chains and Scenarios.	SA3 Define System Functions. Define Functional Exchanges and components	SA4 Allocate System Functions to System and Actors	M&S-SA5 Define system modes and states	D-SA6 Define system data model	I-SA7 Define interfaces and describe interfaces scenarios Enrich Logical Scenarios.
Logical Architecture	R-LA Derive system requirements and capture components requirements	LA1 Transition Capabilities Realization from system layer	LA2 Define Functional Chains and scenarios	LA3 Derive System Functions and define Logical Functions. Define Functional Exchanges and components.	LA4 Allocate Logical Functions to Logical Components	M&S-LA5 Define logical components modes and states	D-LA6 Define logical data model	I-LA7 Delegate System Interfaces and create Logical Interfaces. Enrich Logical Scenarios.
Physical Architecture	R-PA Derive logical requirements and capture physical requirements	PA1 Transition Capabilities Realization from logical layer	PA2 Define Functional Chains, Scenarios, and Physical Path	PA3 Derive Logical Functions and define Physical Functions. Define Functional Exchanges and components.	PA4 Define Physical Nodes and refine Behavioural Physical Components. Allocate Behavioural Components.	M&S-PA5 Define physical nodes modes and states	D-PA6 Define physical data model	I-PA7 Delegate Logical Interfaces and create Physical Interface. Enrich Physical Scenarios.

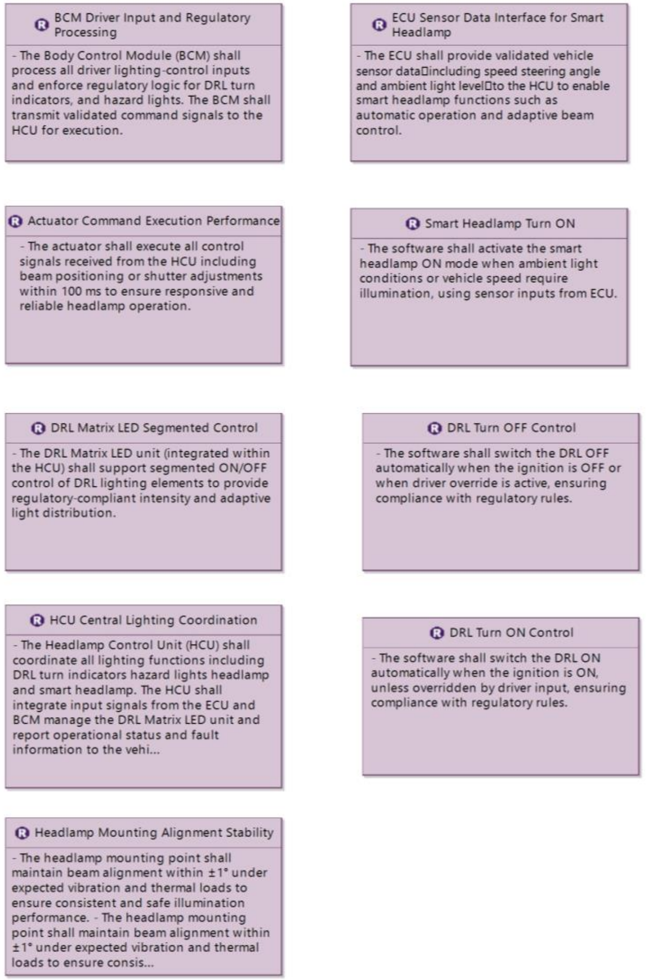
Source: https://mbse-capella.org/

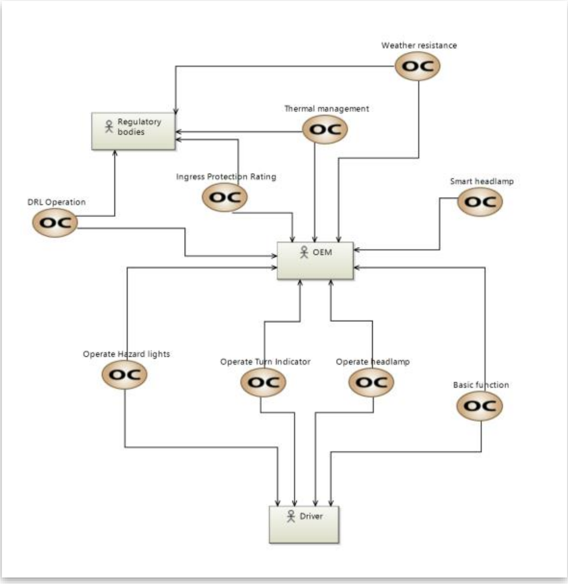
OA LAYER



Operational Entity Diagram

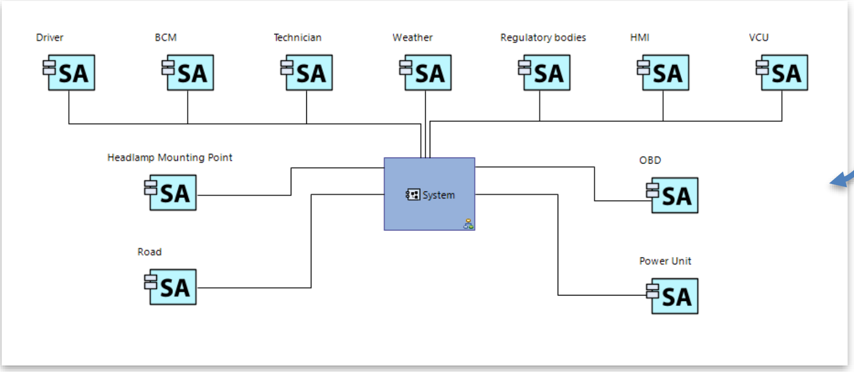
Stakeholder Requirements





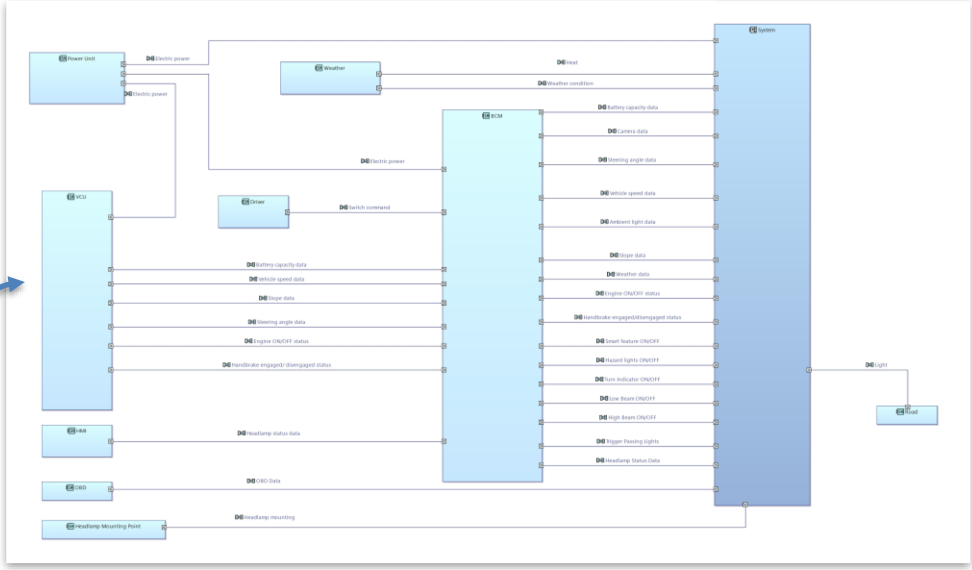
Operational Capability Diagram

SA LAYER

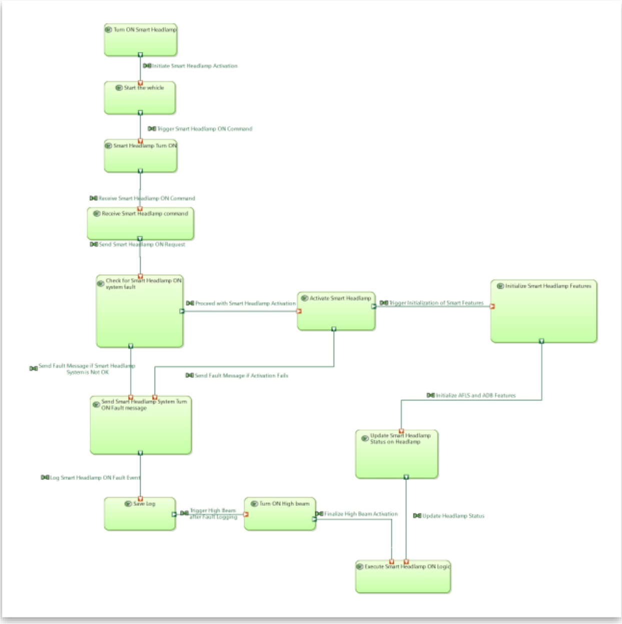


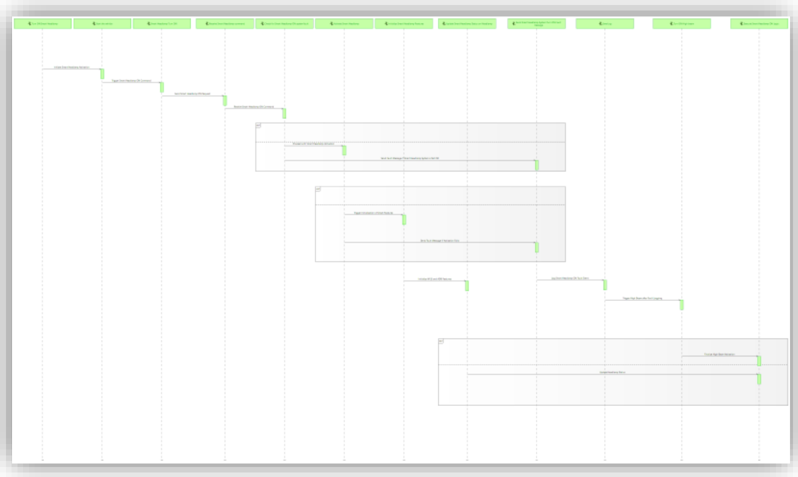
System Contextual Diagram

System Architecture Diagram

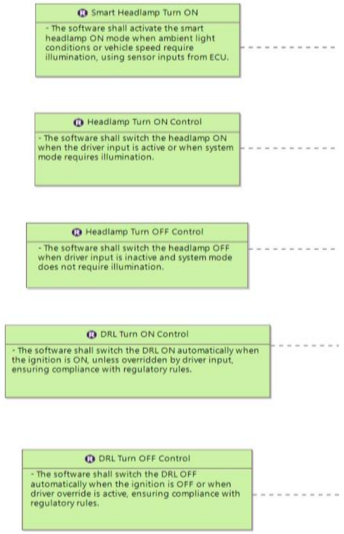


System Data Flow Diagram



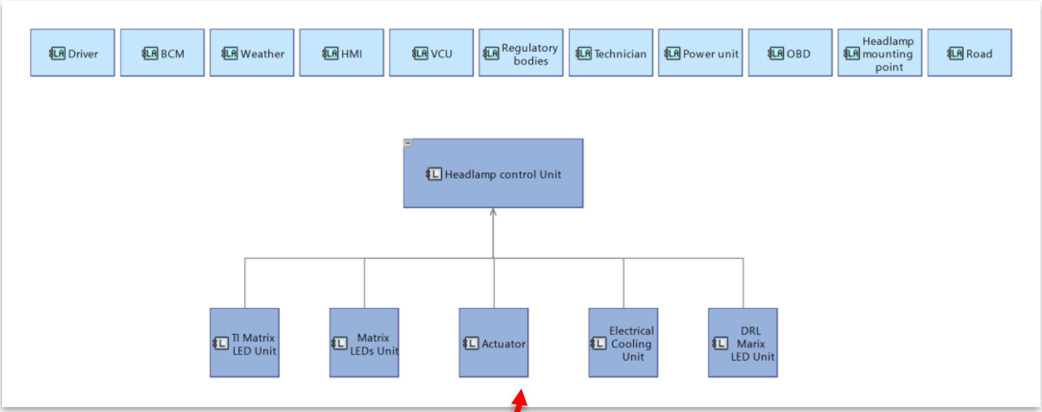


Functional Scenario Diagram



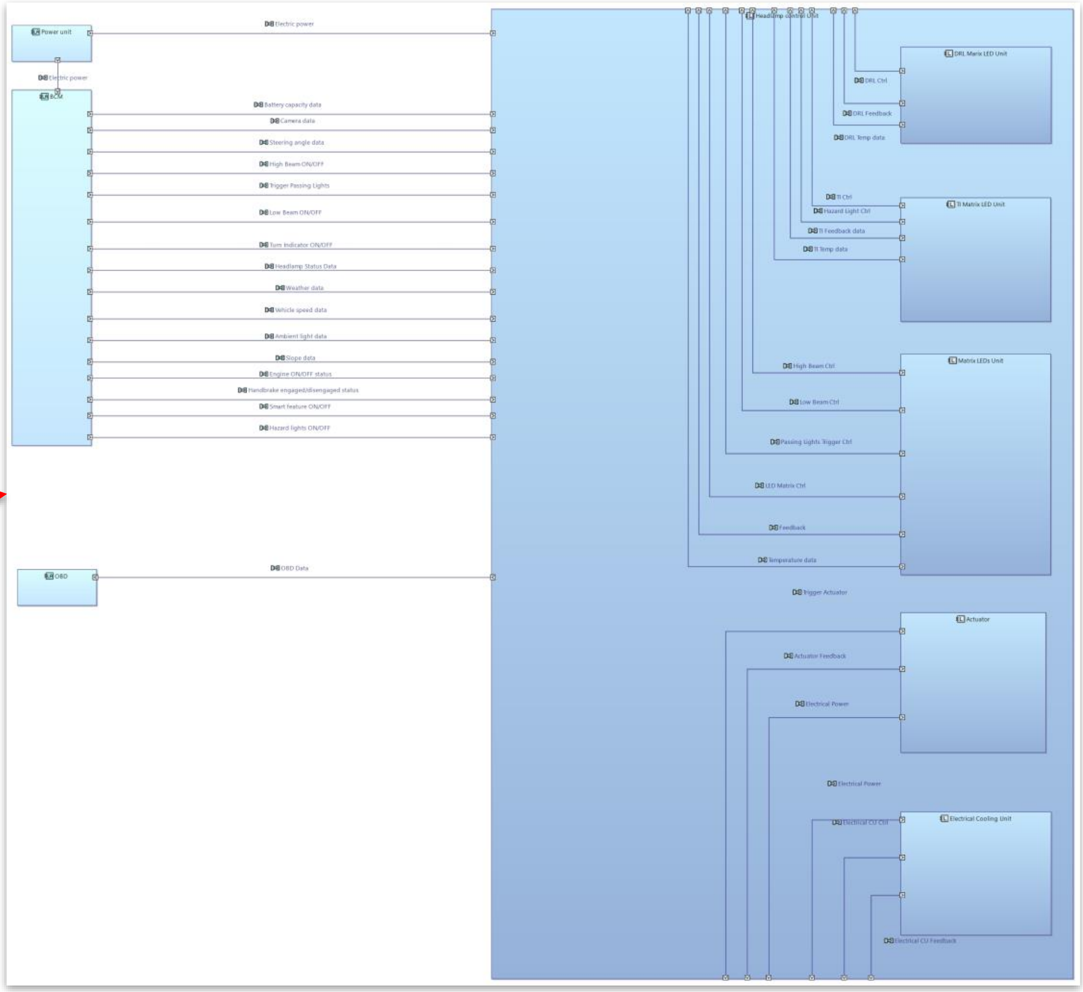
System Requirements Diagram

LA LAYER



Logical Component Breakdown Diagram

Logical Architecture Diagram



Questioning Myself:

Have I produced the work products expected by ASPICE?

Are there any requirements that are still unallocated?

Are all interfaces clearly defined?

Is every element in the LA traceable to a requirement or a function?

Is the Logical Architecture detailed enough to start Software and Hardware Architecture activities?

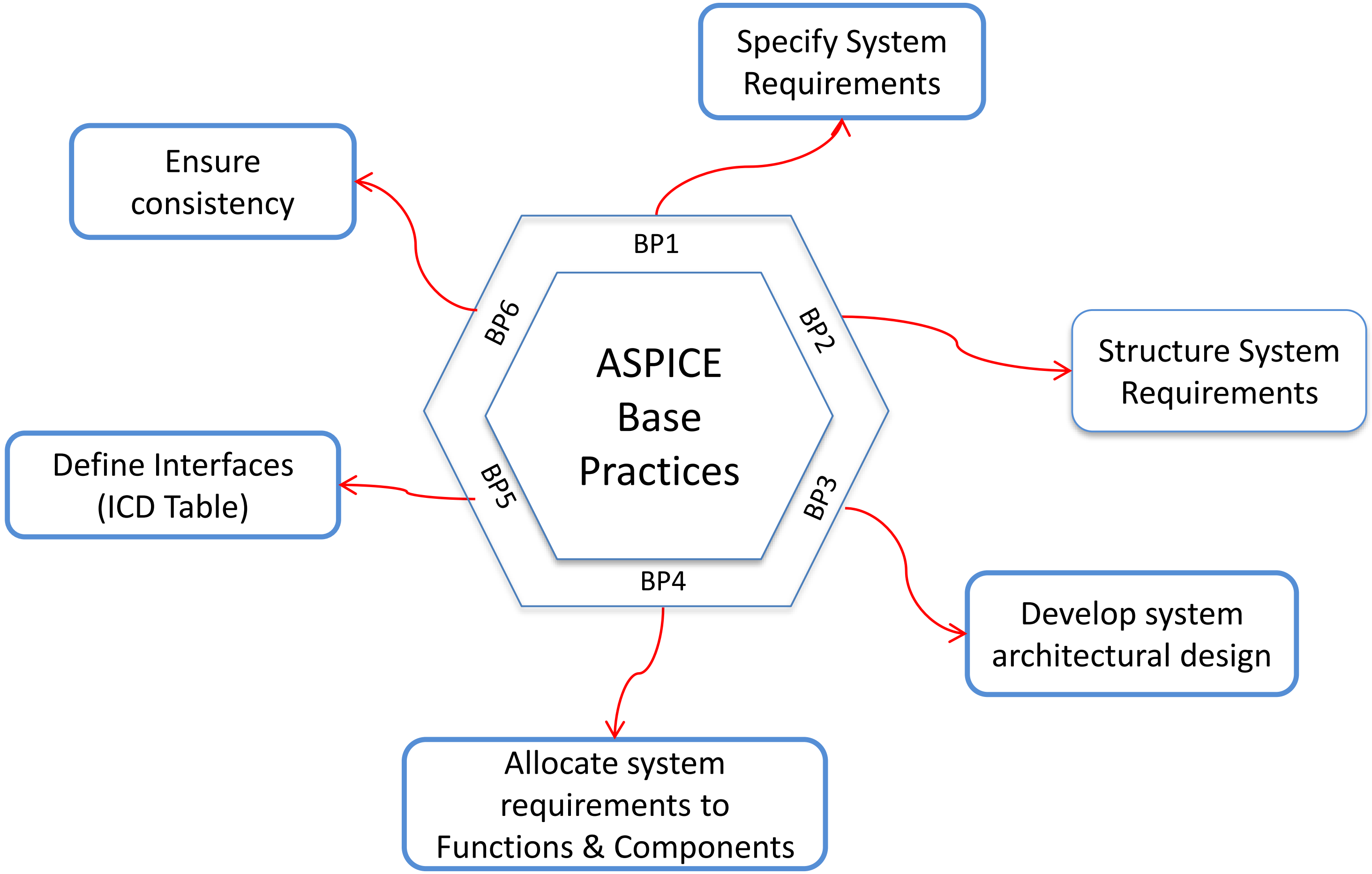
Have I checked the model for inconsistencies or missing links?

Did I validate that the Logical Architecture fulfills all System Requirements?

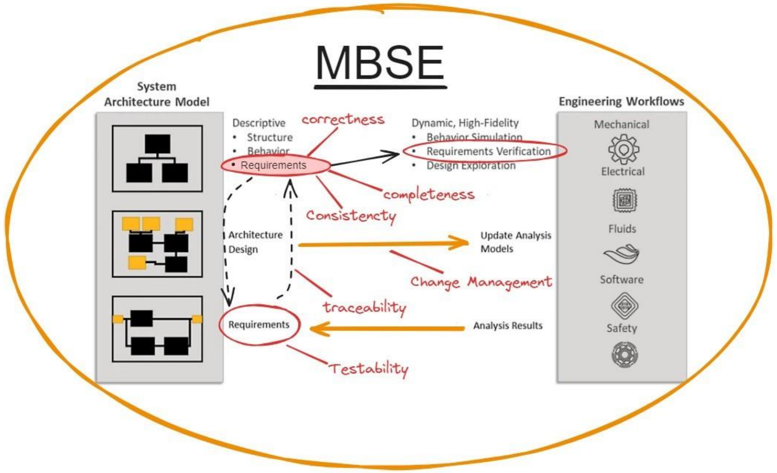
How modelling in Capella benefits us: Using ASPICE Process

ASPICE Capability Levels: (L0-L5)

ASPICE – Assessment Level	Name	Focus
0	Incomplete	Incomplete
1	Performed	The process purpose is fulfilled by executing the base practices and generating output work products
2	Managed	The process performance is planned and monitored at the project level
3	Established	The process performance follows an organization-wide defined standard process
4	Predictable	Process execution is controlled by metrics
5	Innovating	Metrics obtained from process execution are used to optimize processes



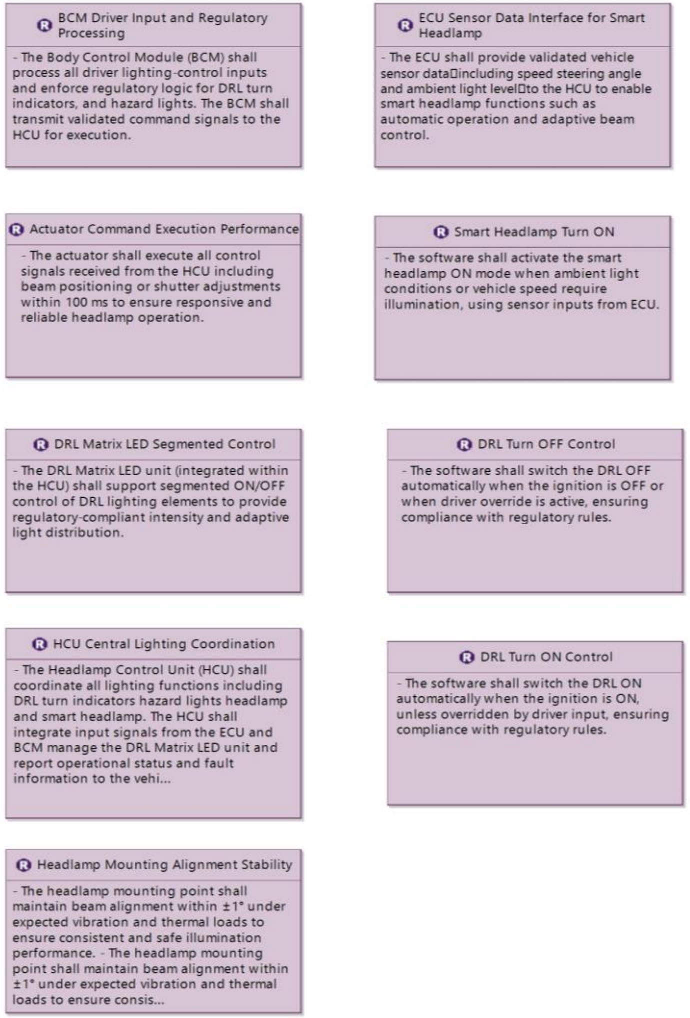
BP1: Specify System Requirements



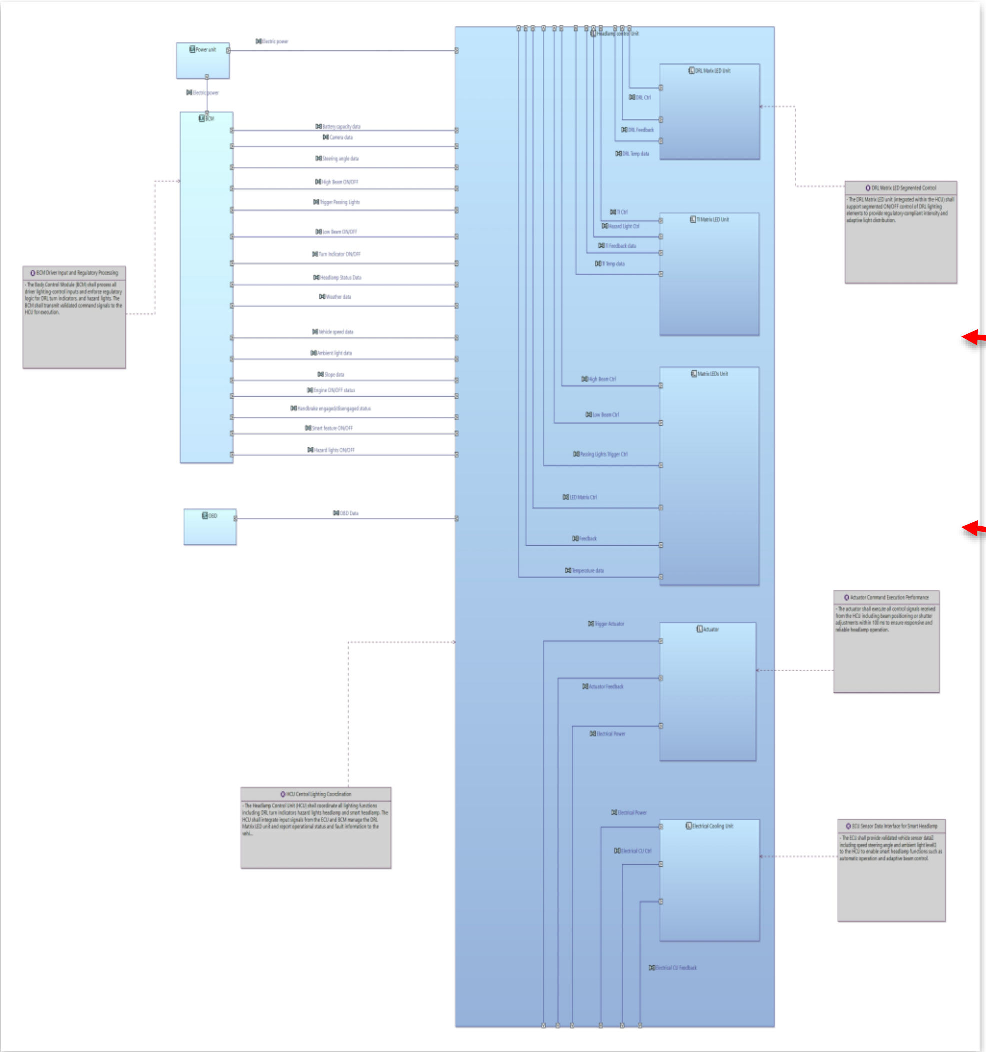
Sourcehttps://www.linkedin.com/pulse/importance-quality-automotive-systems-requirements-context-jorge-qnglf

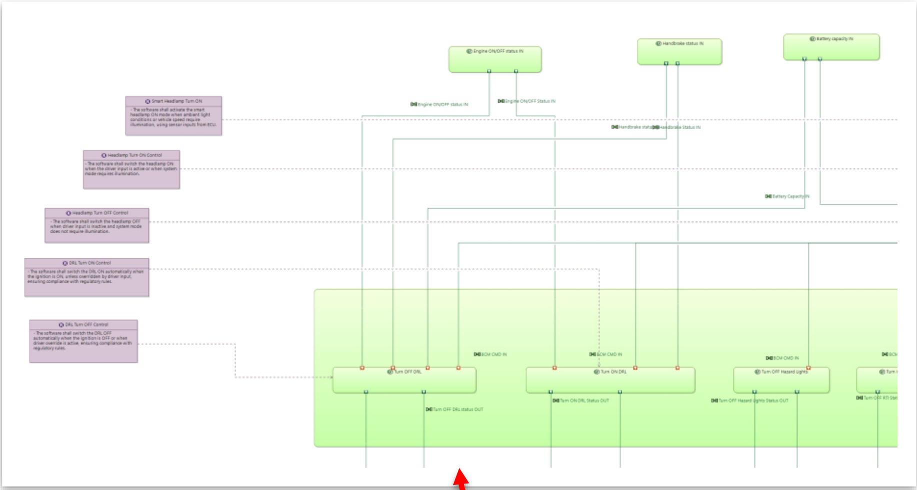
All system requirements (functional & non-functional) are captured using templates

Capell Add-on used
REQUIREMENT VIEWPOINT



BP2: Structure System Requirements

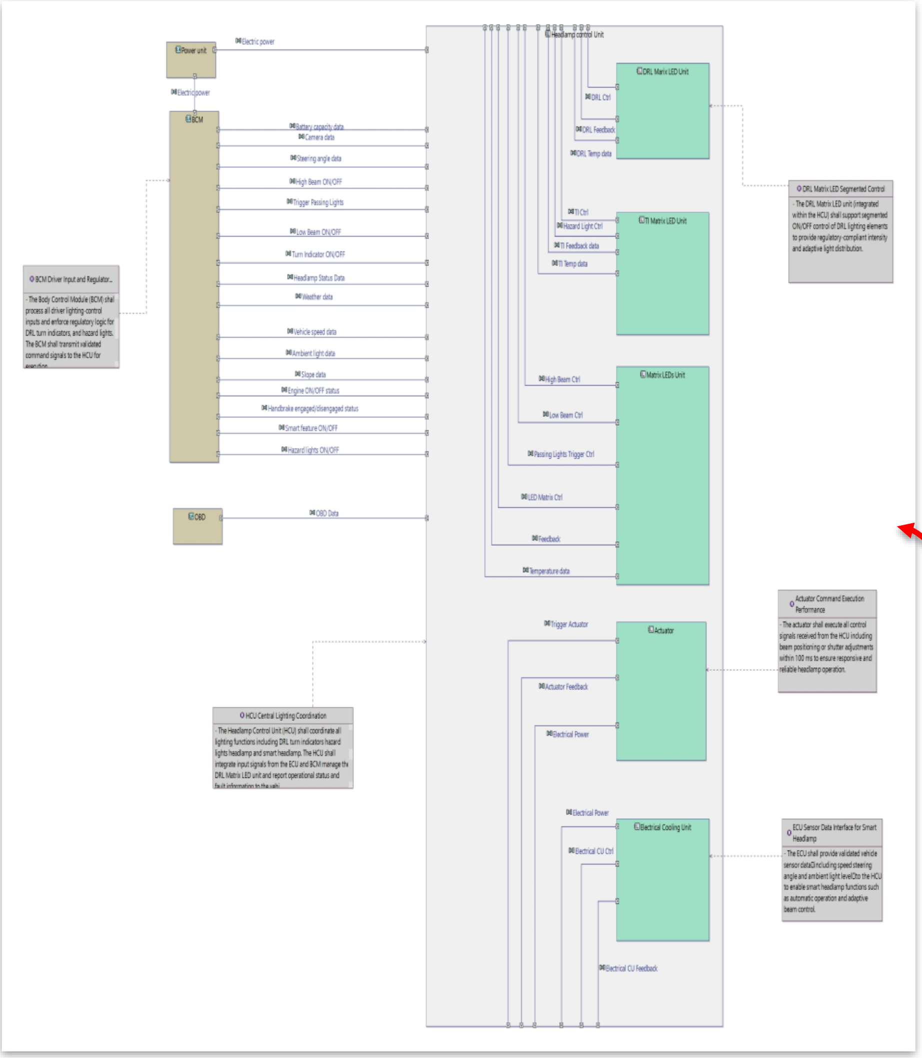


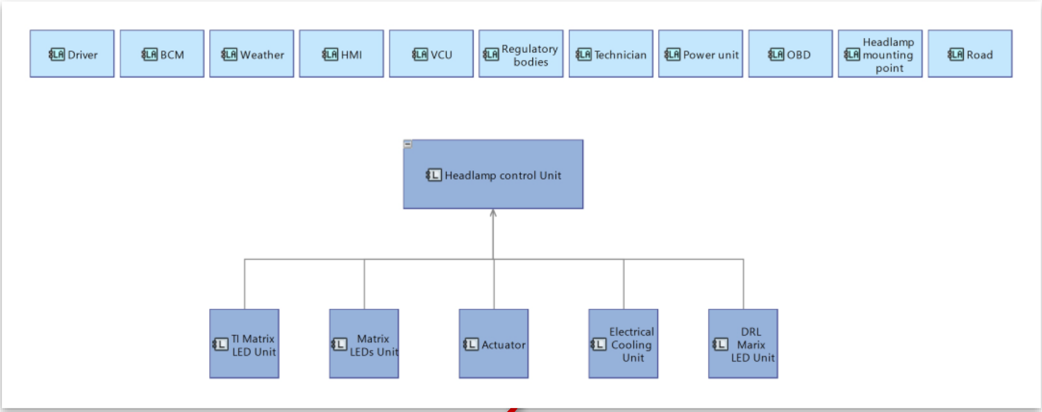


Structured by domain/component/ function

BP:4 Assign system requirements to architecture components.

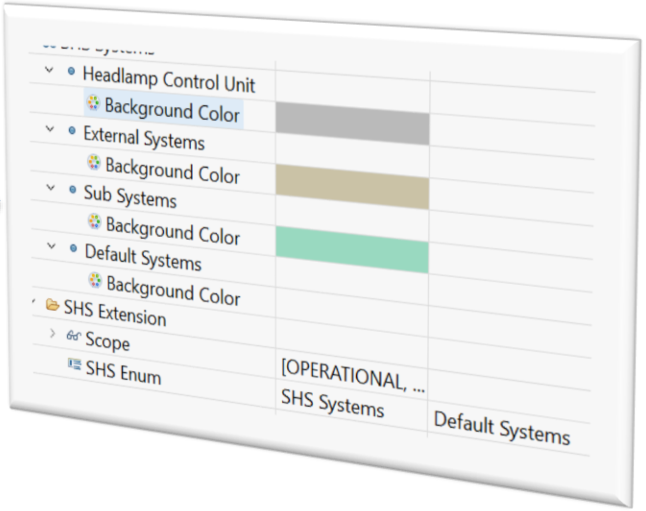
BP3: Develop System Architectural Design





Create an architecture that satisfies system requirements

Capella Add-on used: Diagram Styler



BP5: Define interfaces (ICD Table)

- Python4Capella
  - java_api
  - resources
  - sample_scripts
    - Autoname_Logical_Components_Actors_and_System.py
    - Export_a_pseudo_hierarchy_of_PA_elements_to_xlsx.py
    - Export_capabilities_and_owned_scenarios_to_xlsx.py
    - Export_description_of_elements _with_html format_and_as_plain text_to_xlsx.py
    - Export_list_of_functional_exchanges_to_xlsx .py
    - Export_list_of_Requirements_to_xlsx.py
    - Export_Missions_capabilities_FC_status.py
    - Export_missions_exploited_capabilities_involved_FC_to_xlsx.py
    - Export_physical_paths_and_involved_links_to_xlsx.py
    - Export_Property_Values_associated_to_elements_to_xlsx.py
    - export_selected_element_diagrams.py
    - Export_SF_and_FE_to_xlsx.py
    - Export_table_for_Node_PC_with_summary_PP_and_PL_to_xlsx.py
    - Export_the_list_of_physical_components_to_xlsx.py
    - Export_traceability_matrix_SF_LF_to_xlsx.py
    - Function_allocation_completeness_check.py
    - HTMLDescription_To_Excel_Description.py
    - Import_breakdown_Node_PCs_from_xlsx.py
    - Import_physical_components_from_xlsx.py
    - List_logical_components_in_console.py
    - List_logical_functions_in_console.py
    - Rename_Logical_Functions.py
    - Rename_Ports_Antagonist.py
    - Rename_Ports.py

Export Functional/Logical Exchanges

Exchange ID	Source Component	Component Exchange Name	Target Component
85623550-ffd4-491f-a3fe-e69a2a7e0217	BCM	Vehicle speed data	Headlamp control Unit
59175b98-88a8-4643-aa0b-44b0e96522db	Headlamp control Unit	Electrical Power	Electrical Cooling Unit
a5631dd8-1be6-415d-987d-534590616ea2	BCM	High Beam ON/OFF	Headlamp control Unit
059107f0-5e81-47f2-aa08-0c14879f987e	Headlamp control Unit	TI Feedback data	TI Matrix LED Unit
859c9424-b9c2-4920-a37b-c0a150391573	BCM	Trigger Passing Lights	Headlamp control Unit
7a3189e9-587a-408d-b1b9-9ac082b6daac	Headlamp control Unit	Electrical CU Ctrl	Electrical Cooling Unit
39025ed2-f5f0-463c-9463-b735473f46f4	BCM	High Beam ON/OFF	Headlamp control Unit
d72aaa30-b8f2-4adc-9816-2812c159ac84	BCM	Engine ON/OFF status	Headlamp control Unit
665b0a42-953e-4a3c-8e60-8375c210b022	Headlamp control Unit	DRL Feedback	DRL Marix LED Unit
2bbbcac7-84ec-4460-bd77-d1b274762e96	Headlamp control Unit	DRL Temp data	DRL Marix LED Unit
88ac332f-69d1-4b80-a1ca-70a809241479	Power unit	Electric power	Headlamp control Unit
3ca2e787-0aed-4547-9692-c40ae15b9916	BCM	Camera data	Headlamp control Unit
faa82dde-61d3-454d-8eb4-c00413946685	Headlamp control Unit	DRL Ctrl	DRL Marix LED Unit
249411f2-d707-46e3-b2ba-50dd91ca066c	Headlamp control Unit	Passing Lights Trigger Ctrl	Matrix LEDs Unit
673014ac-be18-420b-831c-2c042bd48448	Power unit	Electric power	BCM
65795b37-ddf0-4a70-aedf-0c311c6dbb75	BCM	Battery capacity data	Headlamp control Unit
eb36855a-c150-41cc-95be-f53b6d7e8240	Headlamp control Unit	TI Temp data	TI Matrix LED Unit
01aa212a-80da-4f64-99da-92891ddce218	Power unit	Electric Power	Headlamp control Unit
2cee1e34-24fb-4857-96ee-5bc0d7a6f050	BCM	Headlamp Status Data	Headlamp control Unit
02bddc4a-3a80-46a9-8616-60666d324591	Headlamp control Unit	Low Beam Ctrl	Matrix LEDs Unit
[illegible]e5a71c-d0fb-467b-8a2c-8e4ca35c462e	BCM	Hazard lights ON/OFF	Headlamp control Unit
9c[illegible]74d4-36ba-4a27-97d3-1e860a978499	BCM	Handbrake engaged/disengaged status	Headlamp control Unit
f4[illegible]32768-1a22-4bc9-872b-605e63d09270	BCM	Steering angle data	Headlamp control Unit
[illegible]2c09472-eeba-4e00-adaf-34431ecc5580	Headlamp control Unit	Hazard Light Ctrl	TI Matrix LED Unit

WHY?

Describe the interfaces between system elements clearly and completely

Capella Add-on used:
PYTHON4CAPELLA

BP5: Define interfaces (ICD Table)

BLUEKEI SOLUTIONS

Checking for duplicates

Checking any errors in Naming convention

Exchange ID	Source Component	Component Exchange Name	Target Component
85623550-ffd4-491f-a3fe-e69a2a7e0217	BCM	Vehicle speed data	Headlamp control Unit
59175b98-88a8-4643-aa0b-44b0e96522db	Headlamp control Unit	Electrical Power	Electrical Cooling Unit
a5631dd8-1be6-415d-987d-534590616ea2	BCM	High Beam ON/OFF	Headlamp control Unit
059107f0-5e81-47f2-aa08-0c14879f987e	Headlamp control Unit	TI Feedback data	TI Matrix LED Unit
859c9424-b9c2-4920-a37b-c0a150391573	BCM	Trigger Passing Lights	Headlamp control Unit
7a3189e9-587a-408d-b1b9-9ac082b6daac	Headlamp control Unit	Electrical CU Ctrl	Electrical Cooling Unit
39025ed2-f5f0-463c-9463-b735473f46f4	BCM	High Beam ON/OFF	Headlamp control Unit
d72aaa30-b8f2-4adc-9816-2812c159ac84	BCM	Engine ON/OFF status	Headlamp control Unit
665b0a42-953e-4a3c-8e60-8375c210b022	Headlamp control Unit	DRL Feedback	DRL Marix LED Unit
2bbbcac7-84ec-4460-bd77-d1b274762e96	Headlamp control Unit	DRL Temp data	DRL Marix LED Unit
88ac332f-69d1-4b80-a1ca-70a809241479	Power unit	Electric power	Headlamp control Unit
3ca2e787-0aed-4547-9692-c40ae15b9916	BCM	Camera data	Headlamp control Unit
faa82dde-61d3-454d-8eb4-c00413946685	Headlamp control Unit	DRL Ctrl	DRL Marix LED Unit
249411f2-d707-46e3-b2ba-50dd91ca066c	Headlamp control Unit	Passing Lights Trigger Ctrl	Matrix LEDs Unit
673014ac-be18-420b-831c-2c042bd48448	Power unit	Electric power	BCM
65795b37-ddf0-4a70-aedf-0c311c6dbb75	BCM	Battery capacity data	Headlamp control Unit
eb36855a-c150-41cc-95be-f53b6d7e8240	Headlamp control Unit	TI Temp data	TI Matrix LED Unit
01aa212a-80da-4f64-99da-92891ddce218	Power unit	Electric Power	Headlamp control Unit
2cee1e34-24fb-4857-96ee-5bc0d7a6f050	BCM	Headlamp Status Data	Headlamp control Unit
02bddc4a-3a80-46a9-8616-60666d324591	Headlamp control Unit	Low Beam Ctrl	Matrix LEDs Unit
b1e5a71c-d0fb-467b-8a2c-8e4ca35c462e	BCM	Hazard lights ON/OFF	Headlamp control Unit
9ccf74d4-36ba-4a27-97d3-1e860a978499	BCM	Handbrake engaged/disengaged status	Headlamp control Unit
f4882768-1a22-4bc9-872b-605e63d09270	BCM	Steering angle data	Headlamp control Unit
72c09472-eeba-4e00-adaf-34431ecc5580	Headlamp control Unit	Hazard Light Ctrl	TI Matrix LED Unit

Relationship Matrix

- State Machine and Capability Function Matrix
- Operational Activities - Requirements
- State Machine and Capability Function Matrix
- System Functions - Requirements
- System Functions - Operational Activities
- System Actors - Operational Actors/Operational Entities
- Interfaces - Capabilities
- Interfaces - Capabilities and Scenarios
- System/Actors - System Functions
- State Machine and Capability Function Matrix
- Logical Functions - Requirements
- Logical Components/Actors - Requirements
- Logical Functions - System Functions
- Logical Components/Actors - Logical Functions
- Logical Architecture Requirement Refinements
- Logical Interfaces - System Interfaces
- Logical Actors - System Actors
- Interfaces - Capabilities
- Interfaces - Capabilities and Scenarios

Component to Function

	Check for system fault	Check beam status	Trigger DRL OFF	DRL turn ON	Update DRL ON	Send DRL Turn ON system fault mes
Headlamp control Unit						
DRL Marix LED Unit			X	X	X	X
TI Matrix LED Unit						
Matrix LEDs Unit						
Actuator						
Electrical Cooling Unit						
Driver						
BCM						
Weather		X				
HMI						
VCU						
Regulatory bodies						
Technician	X					
Power unit						
OBD						
Headlamp mounting point						
Road						

System Function to Logical Function

	Turn ON DRL	Check for system fault	Check beam status	Trigger DRL OFF	DRL turn ON	Update D
Check beam status			X			
Trigger DRL OFF				X		
DRL turn ON					X	
Update DRL ON						X
Send DRL Turn ON system fault message						
Save log						
SystemFunction 9						
Receive Turn OFF command						
Trigger DRL OFF						
Check for DRL OFF System fault						
DRL turn OFF						
Update DRL OFF status on HMI						
Send DRL Turn OFF						
Save log						
SystemFunction 18						
Receive Turn ON Right TI Command						
Check for RTI System fault						
Send Right Indicator system turn ON Fault message						
Right TI Turn ON						
Send message on HMI to turn off Hazard lights						
Start Flashing Right TI						
Update RTI working status on HMI						
Save Log RTI ON						

ASPICE: The Best Scalable Path for Designing Smart Front Lighting System

- From assumptions to assurance ✓
- Architecture with intent ✓
- Controlled complexity ✓
- Quality as a process outcome ✓
- Enterprise advantage ✓

ASPICE doesn’t make smart lighting system easier: it makes it ***manageable, measurable, and reliably deliverable*** *through disciplined,* ***Model-Based Systems Engineering***

Questions & Discussion

Thank You