



HAL
open science

Renault Nissan new Software Strategy

Olivier Guetta, Emmanuel Coutenceau, Kazuhiro Ishigami

► **To cite this version:**

Olivier Guetta, Emmanuel Coutenceau, Kazuhiro Ishigami. Renault Nissan new Software Strategy. ERTS 2018, Jan 2018, Toulouse, France. hal-02156078

HAL Id: hal-02156078

<https://hal.science/hal-02156078>

Submitted on 14 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Renault Nissan new Software Strategy

V07

Olivier Guetta, Emmanuel Coutenceau, Kazuhiro Ishigami

Abstract

The paper presents how, in the context of Embedded Software, Renault and Nissan are addressing the new challenges and define the strategy to face these challenges, such as the increase of functional complexity, or the necessity to deliver much quicker innovations at the right level of quality. We show the major levers that we used in order to improve drastically the robustness of software at launch of new vehicles, such as standardization (AUTOSAR, Genivi, Android), new process, platforming, continuous integration and validation, improvement of quality management system, key principles for a new organization. We then focus on the Alliance Software Center that was completely launched in July 2017.

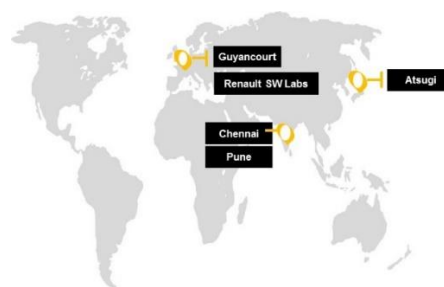
Keywords

Software Development, Software Strategy, Software Process, Software Quality Assurance, Standards, AUTOSAR

1. Introduction

Main challenges for the Alliance are Electric Vehicle, Connected Vehicle and Autonomous Vehicle. In these domains, most of technical innovations rely on Electronic and are realized thanks to Embedded Software. The Alliance wanted to reinforce its development capability in this domain and also improve in a significant manner the robustness of Embedded Software. A breakthrough plan called “Software Robustness Plan” was therefore launched end of 2015.

One outcome of this plan was our willingness to increase our competencies in the domain and modify our internal organization. A new “Alliance Software Center” organization was created, regrouping Renault software resources on one side, Nissan software resources, as well as a new entity called Renault Software Labs, regrouping all Software teams belonging to ex Intel R&D centers located in Toulouse and Sophia Antipolis



The paper is organized in the following manner:

- Recall of Automotive context and Software context
- Description of the 6 main levers of Software Robustness Plan
- Missions and expectations of our new Alliance Software Organization

2. Automotive and Software Context

Functional complexity in vehicle is growing rapidly, Software becomes a key lever for innovations since more and more innovations are realized with electronic and Embedded Software. Embedded Software is therefore increasing exponentially in terms of number of functions and number of lines of code. This key role of Software inside our vehicles will continue to increase with connected and autonomous vehicles. In addition, we need to accelerate Time to Market, improve Software Quality in our vehicles and start managing Updates Over the Air for all embedded software.

We came to the conclusion that a real breakthrough in our embedded software development process was compulsory and that Renault Nissan should really master the development of these software and take into account the following main stakes and levers:

- Reduce SW development cost, thanks to:
 - Increase of Software Components reuse
 - Adoption of Platforming logic with a main Software Product line and re-usable applications (no redevelopment of existing features)
 - Reduction of supplier D&D cost (Design & Development) related to SW development, definition of purchasing strategy and target on supplier D&D, better challenge of development costs
- Improve SW Robustness, described in the next chapter, with a target of reducing the number of SW bugs found by our customers at the launch of a new vehicle

- Become responsive to market, accelerate time to market, for instance by being able to adapt ADAS (Advanced Driving Assistance Systems) functions to a new vehicle quickly, or provide regularly updates of SW (over the air) for bug fixing and new functionalities.

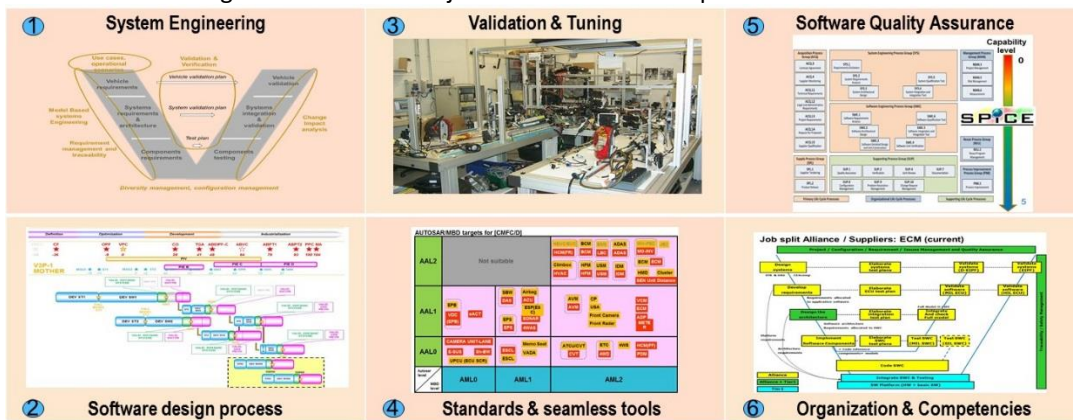
In addition, most of our competitors have already started a transformation in this domain, either by hiring new people or acquiring new companies, and have also launched new methods for their embedded software development (Off cycle activities, new types of architecture to ease reuse, enhanced validation, continuous integration, ...).

3. Description of Software robustness Plan

Software Robustness Plan is a breakthrough plan to improve robustness of Software, including all Renault & Nissan Engineering, Quality and Purchasing divisions developing Systems and Software, and also including all suppliers. Robust Software is defined by:

- Requirements fully defined and SW Product fully compliant
- + Tolerance to faults, tolerance to environment evolutions (for example external driving conditions, or temporary absence of an information, ...)
- + No bug found in serial life by final customer ... even though 0 bug for such SW and such level of complexity is a very hard target (almost not feasible!)

Software Robustness Plan is organized around 6 major levers shown in the picture below:



All levers are important and need to be worked in parallel to ensure robustness of SW.

Here are some explanations on the content of each lever and the results already obtained and in deployment as well as remaining work to do.

Lever 1: System Engineering

Target consists in emission of complete, robust, traceable specifications from Systems to Software, refining all stakeholders' needs at ECU level. The objective of this lever is not to deploy System Engineering on all systems, but to ensure from the ECU point of view that all requirements coming from various systems have been reviewed, understood and validated, and that refinement of these requirements becomes feasible and that V&V (Verification & Validation) plan can be defined.

This lever is an answer to missing or incomplete requirements, late finding of "embeddability" issues, and late findings of intersystem issues.

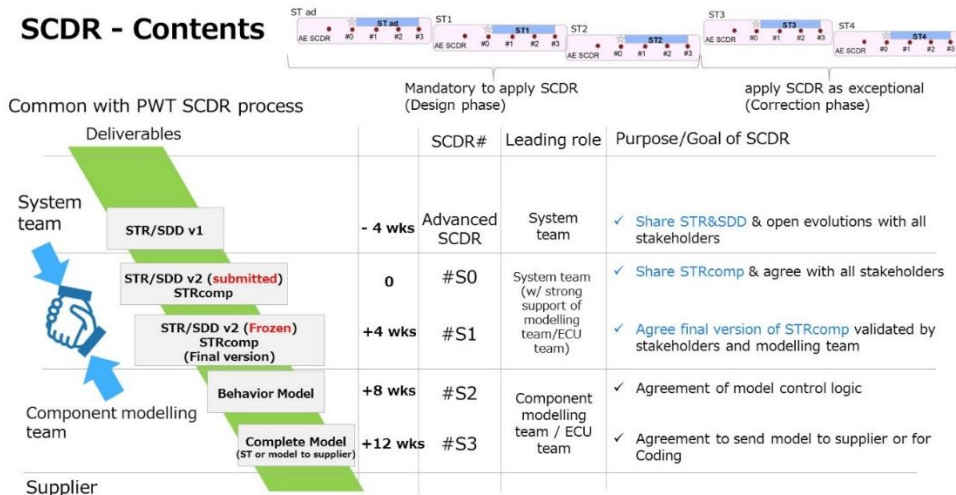
One main action already decided and implemented is the "System Control Design Review" named SCDR.

SCDR is a set of reviews at different milestones of an ECU project that mainly aim at:

- Sharing and clarifying system and then software requirements (including specific safety requirements) with all stakeholders
- Agreeing on the use cases and test plans
- Ensuring traceability between requirements and models in case of function modelling

SCDR process has already been applied in Powertrain division and guarantees a better understanding of the requirements without additional resources.

SCDR - Contents



STR = System Technical Requirements; STRComp = System Technical Requirements for components
SDD = System Design Document

Lever 2: Software Design Process

Target consists in definition of a dedicated process for all embedded software development, taking into account all activities independently whether they are performed by Renault people or Tiers 1 (or even Tiers n). We usually differentiate:

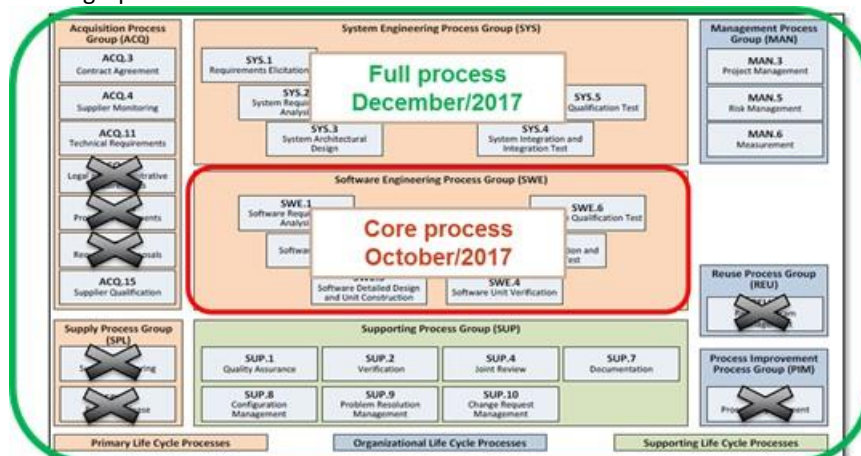
- “black box ECUs”, for which we deliver written high level specifications, and the chosen Tier1 is in charge of providing Hardware and Software
- “white box ECUs”, for which we provide part or total of low level specifications that have been modelled and validated internally. On some cases, we provide a Matlab / Simulink model which is an implementation model and on which the Tier1 can realize the code using an automatic code generator.

The process shall also comply with Renault / Nissan main milestones of Vehicle and Powertrain developments.

Another objective of this lever is to ensure availability and robustness of specifications sent to our Tiers1, and traceability to system requirements.

Last but not least, software updates are costly and usually occurs too long after launching of the vehicle. Therefore, current strategy of managing software corrections and evolutions will have to change and take benefit of « Over The Air » updates.

The main outcome of this lever is the dedicated SW process for which a first and almost complete version will be made available end of December 2017 for full application starting on some ECUs in January 2018. We already know that some evolutions will occur in a version 2 scheduled mid 2018, for instance to take into account the Automotive Spice process marked with a cross on the graph below.



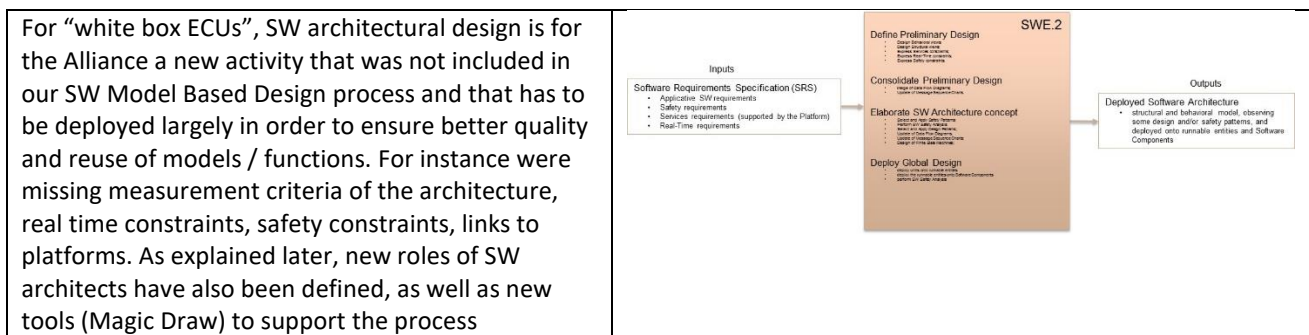
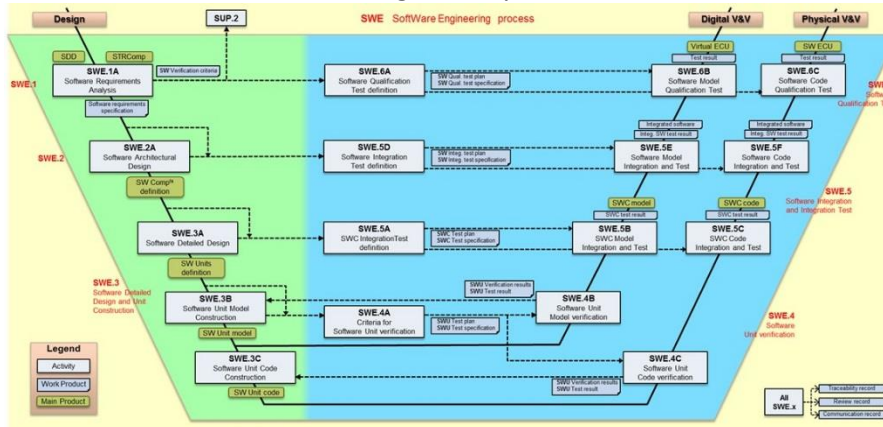
The process relies on Automotive Spice maturity model (some process areas have been removed for the first version) and also takes into account best practices already in use in some teams within the Alliance.

The description of the process will eventually contains:

- Document showing at high level the “identity” of the process
- Description of macro activities
- Detailed description of sub processes

- Instructions and templates to allow designers to provide the expected deliverables during the development of their SW project

The core process of SW activities was defined along the V Cycle :



Additional sub processes and activities to fully comply with ISO26262 requirements for all ASIL levels functions will be included in a future revision of the process already scheduled mid 2018.

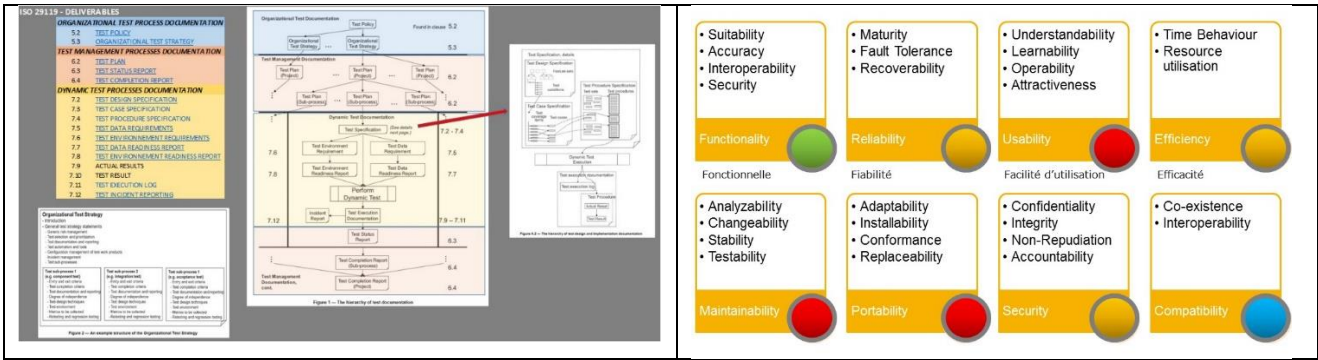
Lever 3: Validation & Tuning

This lever had 3 main objectives:

- Define all the validation activities to be performed, including all associated methods and tools, as a contribution to our SW development process described above. A specific focus was required for calibrations in the Powertrain domain.
- Ensure that validation activities at the software level are done in the respect of QCT (Quality, Cost, Time to Market) project’s commitment, such as:
 - ensure that all requirements allocated to SW are covered by at least one validation activity
 - ensure that all validation activity scheduled to cover all requirements allocated to SW have been executed and have delivered results.
- Propose new methods and tools for end to end validation of ADAS / AD and connectivity systems, relying mostly on simulation, and taking into account safety and security issues. This part will not be shown in this article.

The 3 main deliverables of this lever are:

- Full analysis of current SW validation process, documents, tools per design team. This picture showed a huge diversity among teams, but was also an input to the target process in terms of extracting the best practices and export them to all the teams. The need to improve the quality of specifications and of the test plan was identified, as well as the problem of availability of validation resources.
- Contribution to SW process described above, with inputs taken both from Automotive Spice (Software Process Improvement and Capability determination), CMMi (Capability Maturity Model Integration), ISO 29119 (Software Testing Standard) and ISO 26262 (international standard for functional safety of electrical and/or electronic systems in production), ISTQB (International Software Testing Qualifications Board):

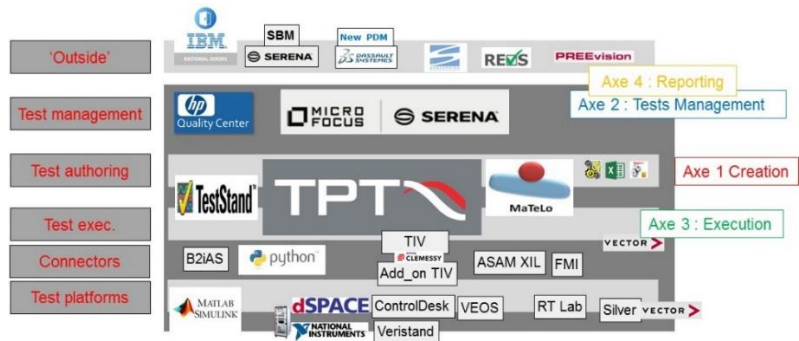


- Selection of what type of validations shall be performed when and on what validation test facilities:

- VALIDATION LEVELS
- V CYCLE- VALIDATIONS
- TEST EXECUTION TARGETS
- MIL SWC
- SIL SWC
- MIL ECU (FULL MODEL)
- MIL (VIRTUAL ECU)
- HIL ECU
- D-EIPF
- EIPF

MIL = Model in The Loop
 SIL = Software In the Loop
 HIL = Hardware in The Loop
 SWC = Software Components
 (D)EIPF = (Digital) Electronic Integration Platform

- Definition of a Proof Of Concept (POC) study on various tools (test creation, test management tool, test script tool, test reporting tool, test bench execution tool) that will provide the generation of the expected documents. The results of the POC will be obtained beginning of 2018.



Some activities still have to be continued, such as:

- Choice of tool set for the management of validation activities (at the end of the POC mentioned just above) and then deployment of these tools in a transversal manner
- Finalization of Continuous integration process definition and integration in the new Software Design Process described above
- Evolution of validation process to fully comply to ISO26262 requirements
- Links with configuration management and issue management

Lever 4: Standards & seamless tools

This lever aims at deploying as much as possible:

- international and widely used standards and practices, such as AUTOSAR architecture and methodology, Genivi or Android middleware platform for Multimedia systems, Model Based design (MBD) methodology at SW level,
- standard, widely used and seamless tools supporting all the processes and easing all exchanges both internally and with our suppliers.

Deployment of MBD SW on “white box ECUs” started a few years ago as an advanced engineering methodology project, already Renault Nissan transversal, already involving entities in both Vehicle and Powertrain Divisions.

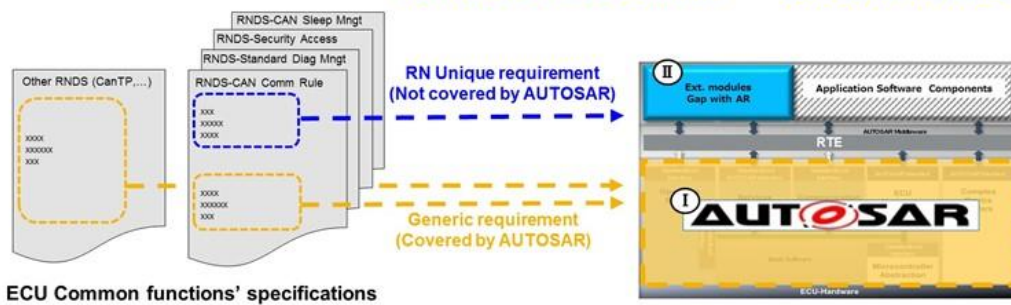
The main deliverables of this lever are:

- A shared definition of AML (alliance Modeling level) and AAL (Alliance AUTOSAR level), as illustrated below

Alliance MBD Level Definition			Alliance AUTOSAR Level Definition		
AAL2	No Models Models are used to validate requirements. The requirements are used for coding. Criteria: - R&N can check validity of requirements by Model-simulation.	Models are used for coding. Criteria: - R&N can create auto-codeable application models.	AAL2 Alliance AUTOSAR Level 2	R&N benefit: - Integration cost reduction and quality improvement by re-using auto-codeable application models with different suppliers (AUTOSAR platform). - re-using standardized and unambiguous software-interface definition complying with AUTOSAR (Arxm) using basic-software services of Autosar platforms). OEM implements auto-codeable application models, whose interface are compliant to connect on AUTOSAR platform. Criteria: Following 3 criteria are all satisfied. (i) Supplier uses BSW with AUTOSAR 4.x. (ii) OEM create ARXM (ECU Level) with AUTOSAR 4.x. (iii) OEM create auto-codeable model of pure application (without containing BSW function).	
AAL1			AAL1 Alliance AUTOSAR Level 1	R&N benefit: - Implementation and maintenance cost reduction and quality improvement by AUTOSAR platform. Supplier implements AUTOSAR basic software Criteria: - Supplier is ready to apply AUTOSAR.	
AAL0			AAL0 Alliance AUTOSAR Level 0	No AUTOSAR	
Autosar level			Autosar level		
MBD level			MBD level		
AML0 Alliance MBD Level 0			AML0		
AML1 Alliance MBD Level 1			AML1		
AML2 Alliance MBD Level 2			AML2		

- A roadmap for the deployment of this MBD / AUTOSAR strategy on future vehicles
- The definition of an “AUTOSAR reference Platform” illustrated below, containing a “standard AUTOSAR basic SW”, the RTE interface, and a set of 4 extension modules located in the applicative part above RTE to take into account Alliance specificities not consistent with the standard (mainly communication and diagnosis requirements).

Alliance SW Standard Platform = I . AUTOSAR 4.x Basic Software + II . Extension Modules

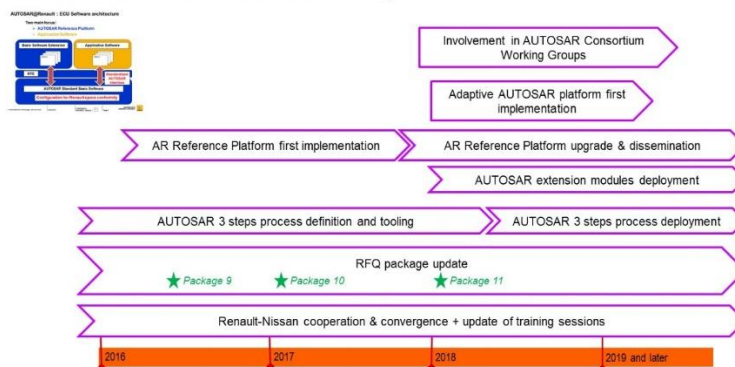


UNIQUE PLATFORM ARCHITECTURE FOR ALLIANCE

- We request our Tiers1 to integrate these extension modules and to consider them as part of the platform

Some activities still have to be completed, the roadmap below has therefore been defined:

AUTOSAR activities roadmap



Lever 5: Software Quality Assurance

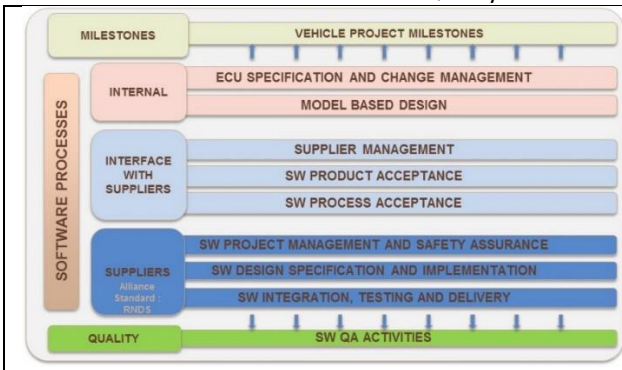
This lever aims at providing an updated Quality Management System able to address all Software Quality and Safety related stakes, both internally and at suppliers’ site.

For many years now, we have been sending requirements to our suppliers, in a specific document for Software called RNDS (Renault Nissan Design Standard). An updated version of this document was expected. In addition, we needed to establish a purchasing policy in order to ensure the right maturity level of suppliers who would be chosen. An extension of the scope was also necessary to cover upstream activities, serial life activities, off board development, and also full capitalization of past software issues for a next generation of expectations.

Furthermore, robustness of software also relies on robustness of deliverables provided by Renault and sent to our suppliers. Application of process and expected outcomes at milestones shall be verified, or coaching shall be provided if activities, methods or tools are unclear.

The main deliverables of this lever are:

- Shared definition of software Quality Assurance activities, as illustrated in the graph below:



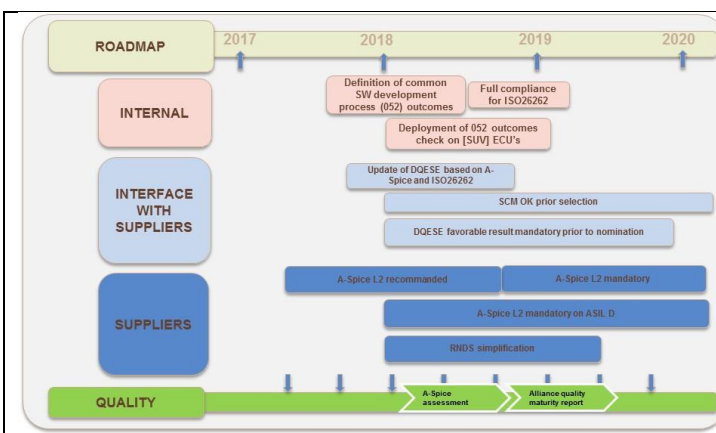
The Main Principle is the conformity of products and processes

A dashboard (quality check) has been defined and is provided at each main milestone of vehicle development, and contains:

- Coverage: all ECUs must be managed with a clear SWQA strategy
- Completeness: All outcomes must be checked
- Consistency: All evidences and status must be evaluated

- The definition and internal deployment of the Dashboard (quality check) mentioned above
- Organization of 2 conventions (in France and in Japan) to explain our new strategy to our major Tiers1.

Some activities still have to be completed, the roadmap below has therefore been defined:



The main milestones of the Roadmap are:

- Check on common internal software development process outcomes
- New common Supplier assessment process (DQESE) bases on A-Spice and ISO26262
- Common Alliance Maturity measurement

SCM = SW Compliance Matrix

Level 6: Organization & competencies

The main objective of this lever is a proposal for a dedicated software organization with both enough resources and resources at the expected level of competencies, to ensure that all activities described in the new process will be performed, and in consistency with the job split of activities between Alliance and Tiers1, that of course may vary depending on the domain and the ECU.

The major outcomes of this lever are:

- A list of “key positions” and associated definition of missions to ensure a robust development, both for System and Software activities, as shown below:

Key positions in System Level		
S.No.	Position	Role
1.	AMS – Metier system architect (Systems Architect)	To contribute to QCDP Engineering performance by committing at VPC on achieving QCDP objectives and by managing global responsibility on the definition, the arbitrage and his or her system integration over the whole product cycle life and projects
2.	PFS - Pilot Functions System (System Functions Leader)	Contribute to product performance and client satisfaction by translating System requirements (customer requirements, regulation, Product General Safety, functional, performance, maintenance requirements...) into technical functions and functional specifications for components
3.	LIS - Leader Engineer System (Engineering Leader)	Ensure product/process development or industrialization according to department rules and specifications adapted to his scope, in compliance with regulations, the legal obligation of the General Product Safety and Renault Design System, and suggest optimizations
4.	AEES	Ensure EE architecture product process convergence (including EE arch validation)

Key positions in SW Development			
#	Position	Role	Day one
1	Product line SW Program Manager	In the perimeter of his Software line, Responsible for all Software developments	Yes
2	Software Project Leader	In the perimeter of his Software line, Manage Software development team for a given ECU	Yes if several Software projects. If not, product line Software Program manager will do the job
3	Product line Sw Architect	In the perimeter of his Software line, Responsible for Software Architecture	Yes
4	System Software Architect	Off vehicle development cycle, responsible to design on the shelf Software architecture for new components	No
5	Vehicle Software Architect	Responsible for instantiate System Software Architectures into a Software Vehicle Architecture	No
6	ECU Software Architect	Responsible for delivering Software Architecture for a given ECU	Yes if several Software projects. If not, product line Software Architect will do the job
7	Software Platform Engineering Leader	Responsible for platform policy (Synthesis, Roadmap, Partnership mngt, Release Mngt)	Yes at least on AUTOSAR
8	Software Integrator	Responsible to assemble the different components to build the Software and contribute to validate Software Architecture	No

#	Position	Role	Day one
9	Software Requirement Analyst	Creating software requirements from other requirements(System, EE archi, Hardware, etc...)	Yes
10	Software Safety Analyst	Support Software Architecture activity taking into consideration safety constraints	Yes
11	Software Modeling Engineering leader	Leading modeling activity and review SWC models	Yes
12	Software Modelling Designer	Design and creating software components	Yes
13	Software Coding Designer	Responsible for code production	Yes
14	Software Validation Architect	Responsible for Validation strategy	Yes
15	Software Validation Engineering Leader	Creating validation plan and management of validation	Yes
16	Software Validation Analyst	Validation software according to validation plan	Yes

Key positions in Supporting team		
S.No.	Position	Role
1.	Vehicle Synthesis SW Engineer (VSSE)	Management of all embedded software development on vehicle project
2	Vehicle SW Quality Engineer(VSQE)	Leading Software Quality Assurance on project
3	Software Quality Engineer (SQE)	Support and Assurance of Software Quality / supplier
4	Software Safety Engineer (SSE)	Support and Assurance of Software Quality / safety
5	Software Multimedia Engineer	Support and Assurance of Software Quality / Multimedia
6	Software MBD Engineer	Support and Assurance of Software Quality / MBD

Some of the key positions described above and their associated activities are “new” for Renault, such as Product Line SW Manager or Vehicle Synthesis SW Engineer.

- A picture of all the “white box ECUs” in one Renault vehicle currently in development, showing adequacy between existing resources on the projects and target key positions defined. Action plans have been proposed (new recruitment, subcontracting, transfer of activities in off shore technical center, ...) to fill the gaps.
- The basics for a new organization that will be described in the chapter below, that lead to the launch of the Alliance Software Center.

4. Missions and expectations of our Alliance Software Organization

In December 2015, Lever 6 produced and shared a diagnosis of embedded software situation in the Alliance. It showed that Alliance had not all in place to take benefit of all the deliverables and proposals from Software Robustness levers toward development efficiency, robustness and value creation.

The decision was taken to identify key basic principles to build an “Alliance Software Center”.

Six principles were defined, shared, and agreed:

ALLIANCE SOFTWARE ORGANIZATION BASED ON 6 PRINCIPLES

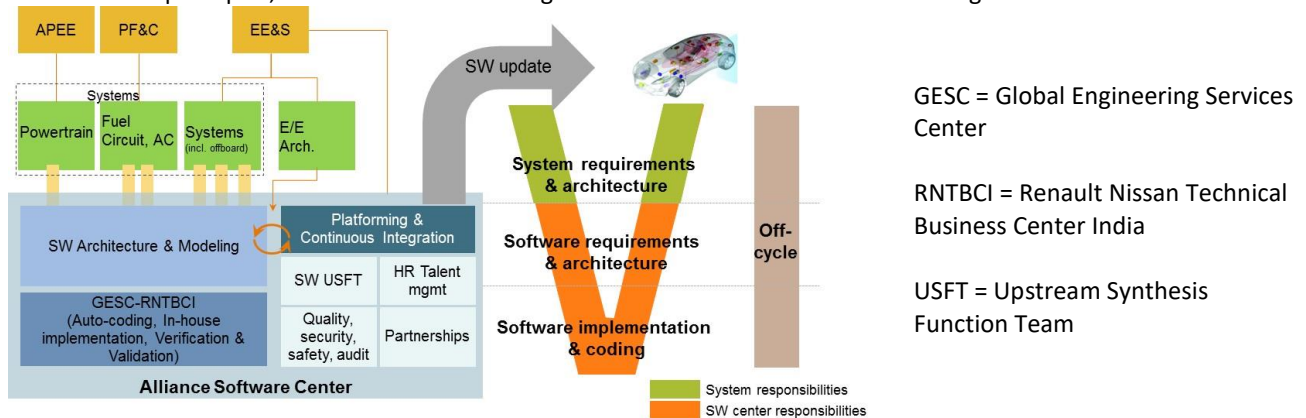


- Critical mass organization: Development of Software is a complex matter including a lot of activities and requiring various competencies. Therefore, from the beginning it was clear that the future organization must own a strong workforce, meaning having enough resources with required level of competency to ensure all these activities. Furthermore, creating a new dedicated organization for Software activity is a clear message to

recruit software talents, showing that software becomes an activity in the Alliance, and that career path and opportunities to grow in an Alliance Software Center will be possible.

- Software shared activities: Efficiency of a Software organization is closely linked to the footprint and job sharing. Whatever functional domain addressed by a Software project, lots of activities are common and can be performed by transversal teams.
- Off project development: Vehicle development is quite vertical whilst the strength of software is to share activities horizontally. The idea is to closely associate System Architects and Software Architects independently of future vehicle EE architecture, topology of ECUs and implementation of functions in these ECUs, and have them work (define, develop, validate) together on software components to be put “on the shelf” for future use. We expect to increase the re-use of software components and drastically reduce the cost of development and time to market.
- Reference platform approach: Industry standards exist and are the result of great contribution from the community, reference platforms shall be developed with respect to these standards. Using these reference platforms internally and requiring our supplier to use platforms compatible with our internal reference platforms will help to improve the quality and robustness of the software.
- Software architecture activity: Software architecture is the most important activity in the development. Architects make the synthesis of Functional and non-Functional requirements in order to propose the best solution. Furthermore, this activity provides a justification of the design that has been chosen.
- Strategic partnerships: Software community in Automotive industry and elsewhere creates lot of value, we want to contribute to it and be part of it. It makes no sense to plan 100% of software development in house. In addition, partnership is a way to accelerate competencies acquisition.

Based on these principles, we have defined the targeted scheme for Alliance SW center organization illustrated below:

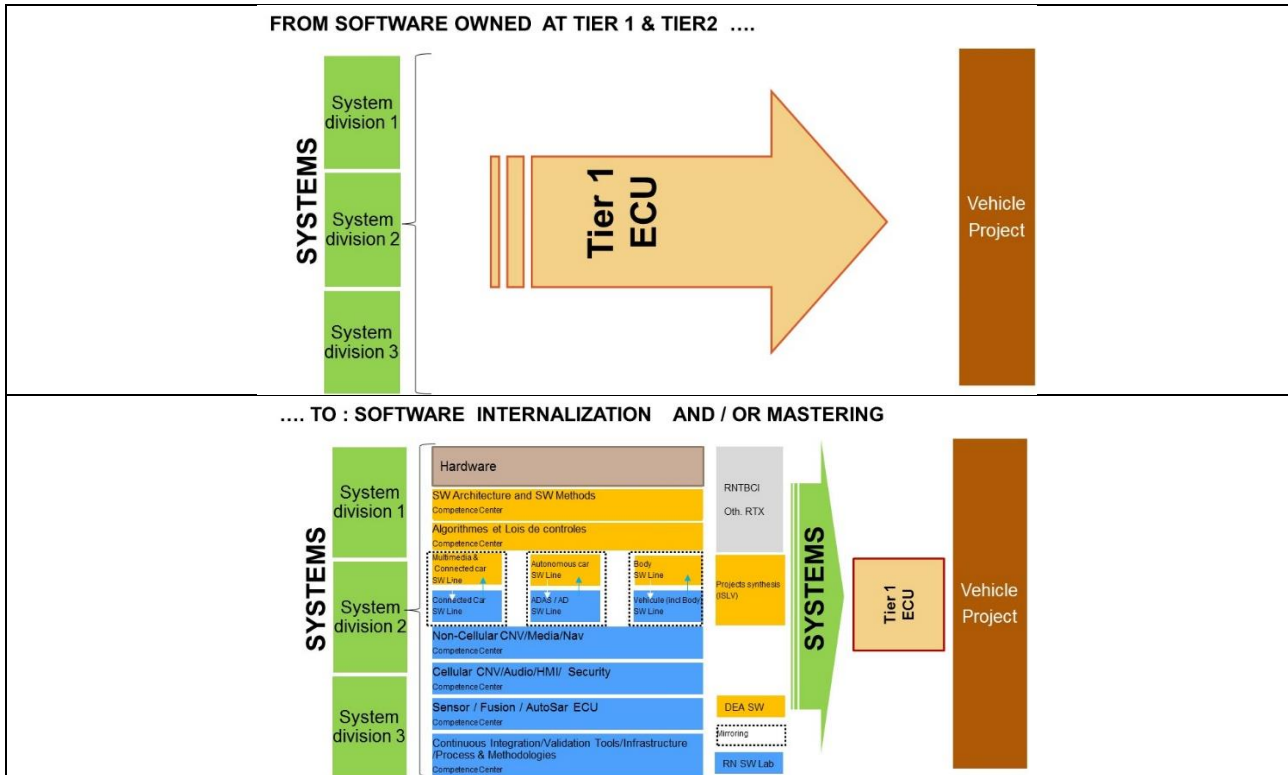


It helped us defining the Nissan and Renault Software organizations and missions.

The Alliance Software Center:

- Is responsible for full vehicle software design, integration, validation, delivery and updates
- Is responsible for algorithms design on systems: ADAS, Chassis Control, Energy management,...
- Owns expertise on SW development
- Owns expertise on Algorithms

In addition, the buyout of the former Intel R&D centers located in Toulouse and Sophia Antipolis, now called Renault Software Labs (RSL), gives us the resources and the competencies we need to accelerate the deployment of our strategy, and in particular the switch from Software developed at Tier1 to Software mastered or developed internally at RSL:



RTx = Renault Technical centers, x being one letter representing the country, such as R for Romania, K for Korea, ...
 RNTBCI = Renault Nissan Technical Business Center India
 CNV = Connectivity (that can be cellular or non-cellular)

The main initial missions of Renault Software Labs are the following:

- Partner with Renault and Nissan to transform the SW company culture and provide SW Internalization & Mastering. Become the SW integration Center for the Alliance
- Responsible for Platforming activities
- Contributor to software “Make vs Buy” decision
- Responsible for definition & implementation of Continuous Integration / Validation across Alliance & partners
- Provide critical additional Competence centers such as Connectivity Cellular /non Cellular, Audio, Media, Security, Algorithms, FOTA (Firmware Over The Air), Architecture, System requirements, Tools.

They will probably evolve in the future !

5. Conclusion

Nissan SW center was launched on April 1st 2017, Renault SW center and Renault Software Labs were launched on July 1st 2017. Detailed missions, ways of working with System Divisions, Governance between all entities, ... have been proposed, as well as (still on-going work) the SW process, methods, tools, templates that all SW centers will have to rely on. Some evolutions are still expected to reach our target, and we will take into account the first return of experiment we will get after a few months with this new alliance organization. We expect to reach within 3 years our target of improvement on SW quality, cost, and time to market.