AUTOCODING WORKING GROUP
Automatic Code Generation for AOCS Flight SW

Davide Oddenino

15th ADCSS, 16/11/2021

Outline of the presentation

- Background – Autocoding Working Group
- Vol I - Process: Autocoding generation process
- Vol. II - Technology: Modelling Guidelines
- Conclusions

Outline of the presentation

- **Background – Autocoding Working Group**
- Vol I - Process: Autocoding generation process
- Vol. II - Technology: Modelling Guidelines
- Conclusions

Autocoding Working Group - Objectives

The purpose of the Autocoding Working Group is to **prepare the ESA Autocoding Handbook** to define for the **AOCS Flight SW Cat. B**:

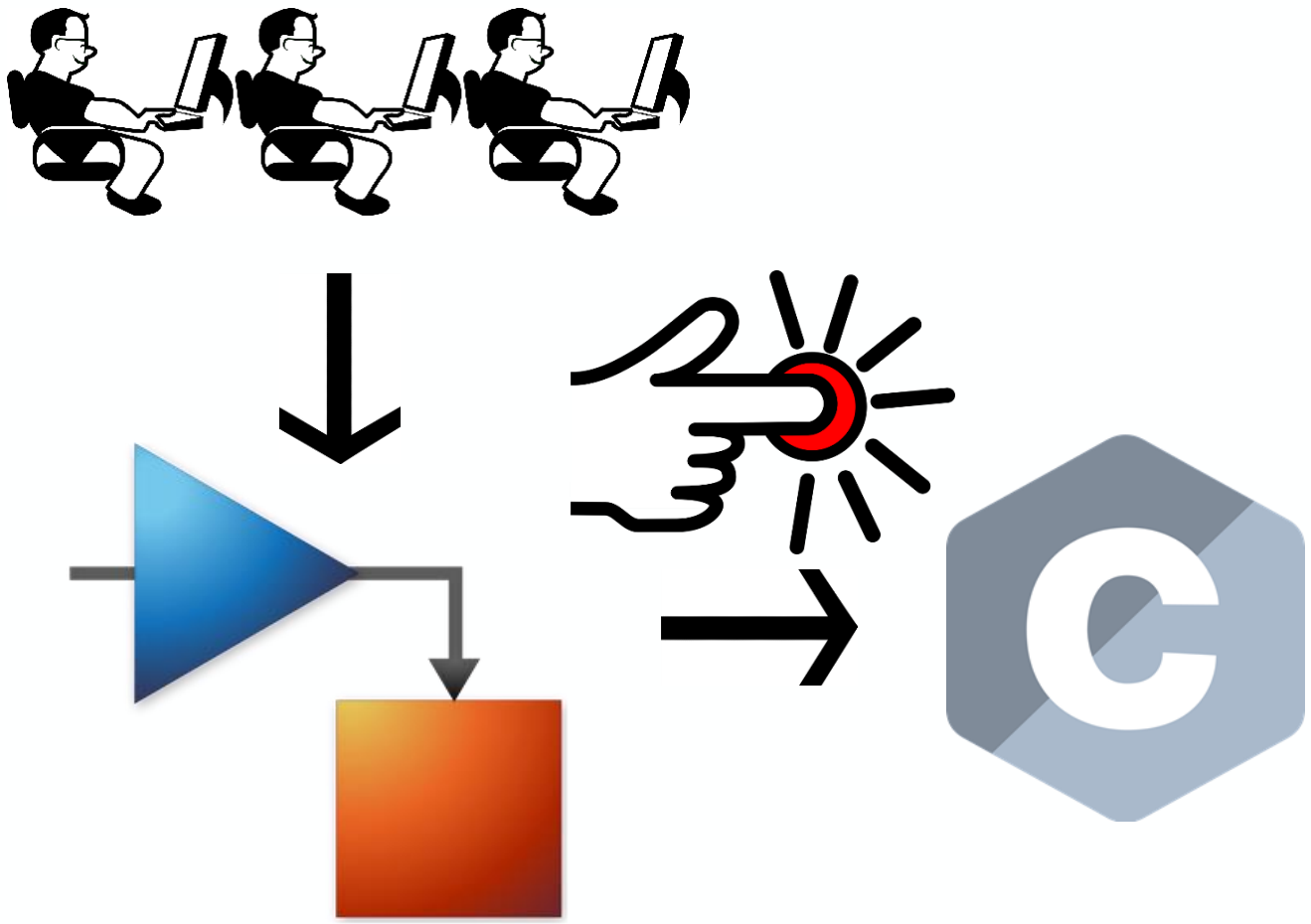
- the **technology** (modelling guidelines, impact of the code generator, etc…)
- the **process** (GNC algorithm development process and application software process covering all the lifecycle up to V&V)

to be used when creating models and generating flight code with the objective to ensure the generated code is **correct, reliable**, readable, sharable/reusable and maintainable.

The intended use of the guidelines are the following ones:

- *Use in **support to projects** providing an harmonized ESA position across the Agency.*
- *Use in **R&D technology** activities.*
- *Promotion of the use of this type of methodology **across the phases of a development**.*
- *Contribution to the **assessment of the quality** of the final software product*

SAVOIR Autocoding Working Group



I. **Vol I development and verification process:**
   - *Verification & Validation process review wrt manual coding*
   - *Automatic steps and available tools (possible improvements, customization) guidelines*
   - *Deliverables (comparison SW dev flow)*
   - *Test reporting guidelines (templates, content description)*

   Compliance with existing standards:
   - *Cross check/mapping of ECSS requirements to the Autocoding V&V process*
   - *Comparison with Autocoding processes as proposed/implemented on different projects*

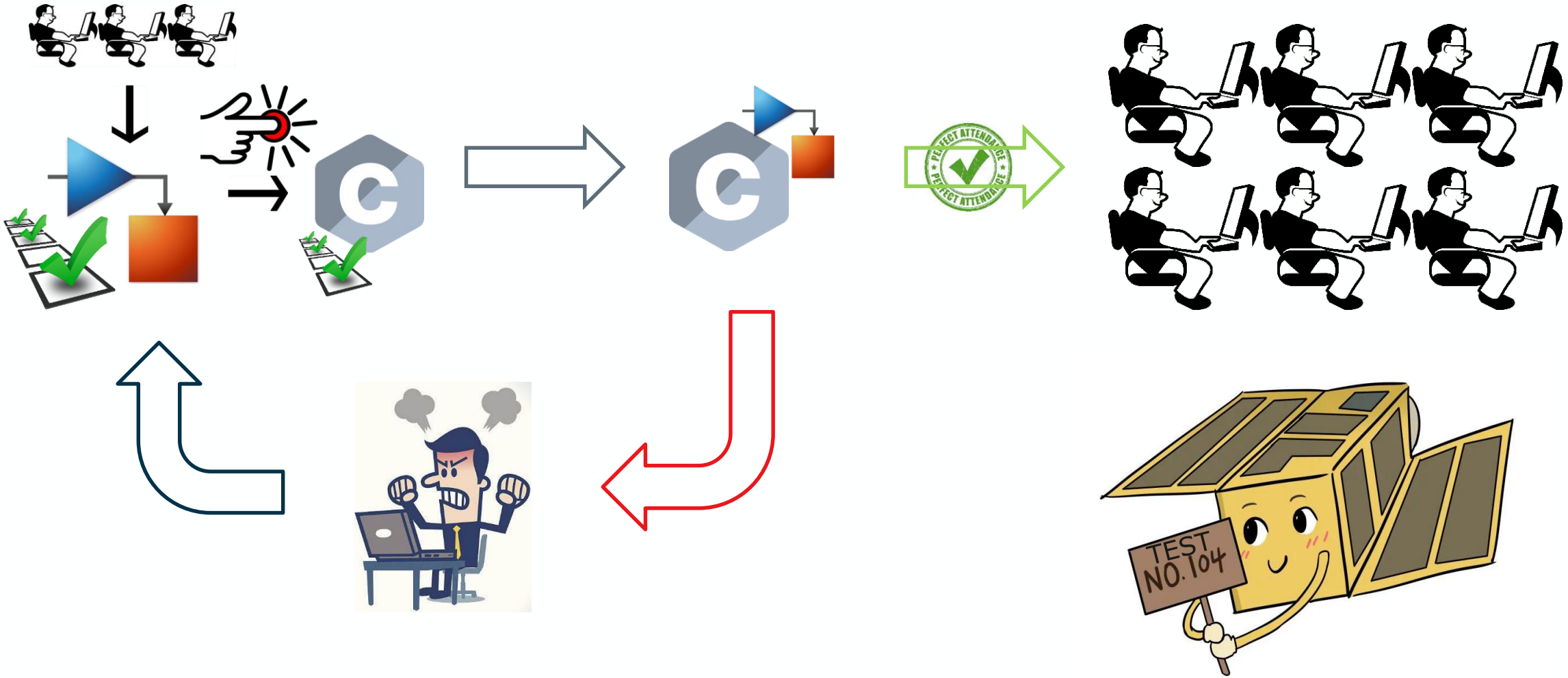
II. **Vol II Modelling guidelines (AOCS modelling):**
   - *Define modelling guidelines*
   - *Configuration of code generation toolboxes*
   - *Classification of guidelines*

SAVOIR AUTOCODE
space avionics open interface architecture

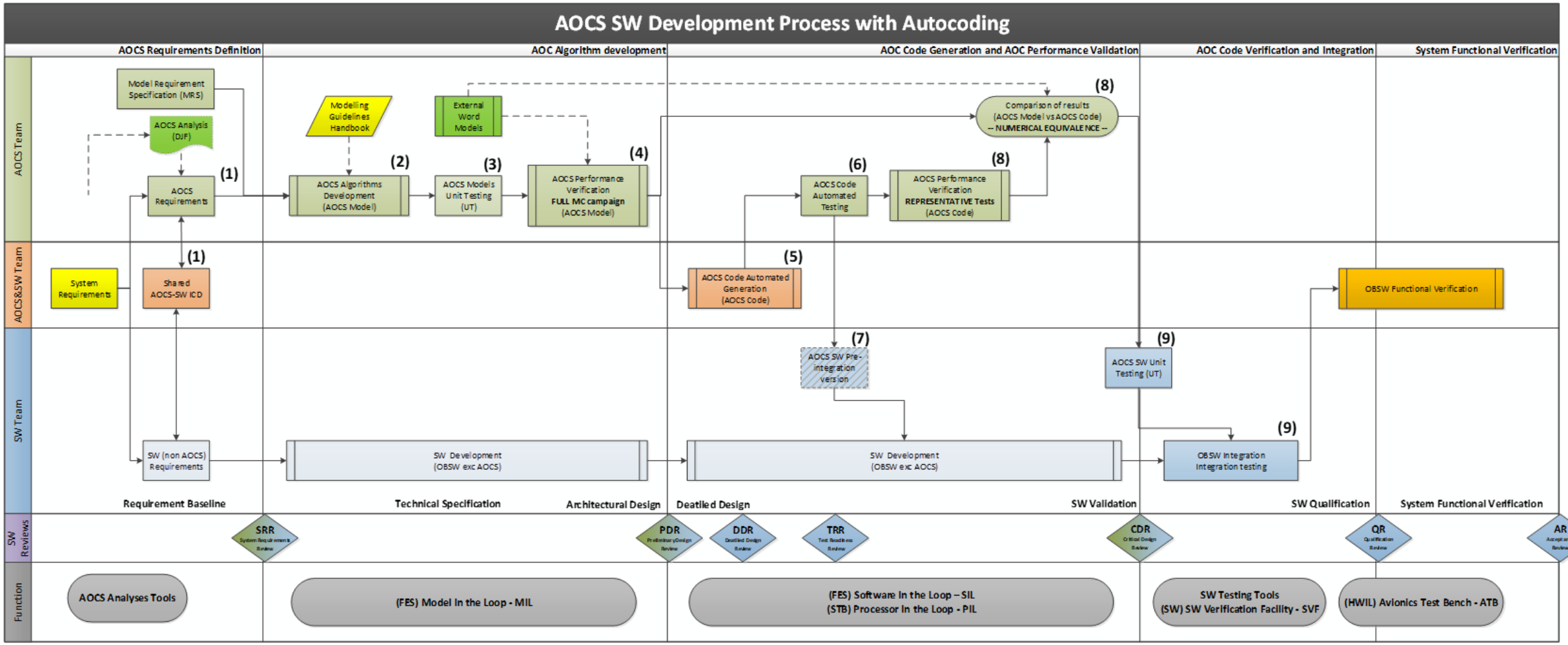
Outline of the presentation

- Background – Autocoding Working Group
- **Vol I - Process: Autocoding generation process**
- Vol. II - Technology: Modelling Guidelines
- Conclusions

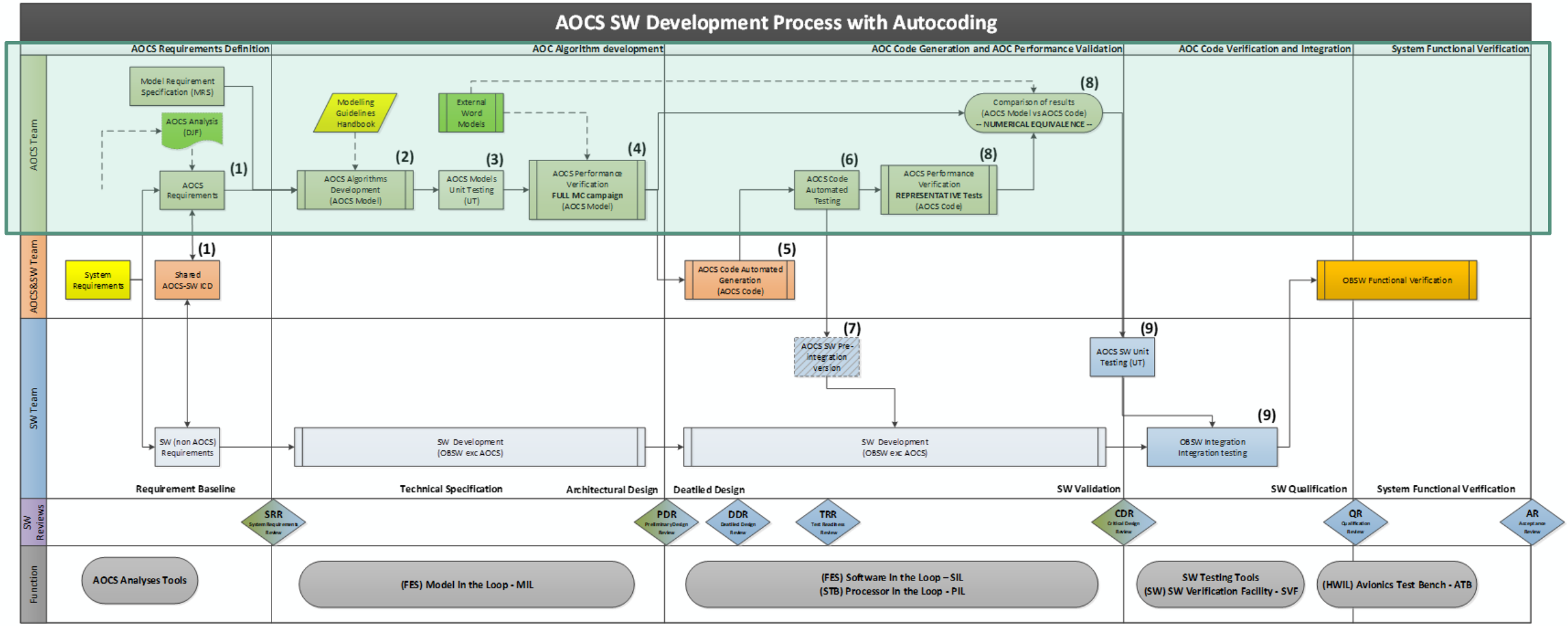
Process - Code generation development & Verification



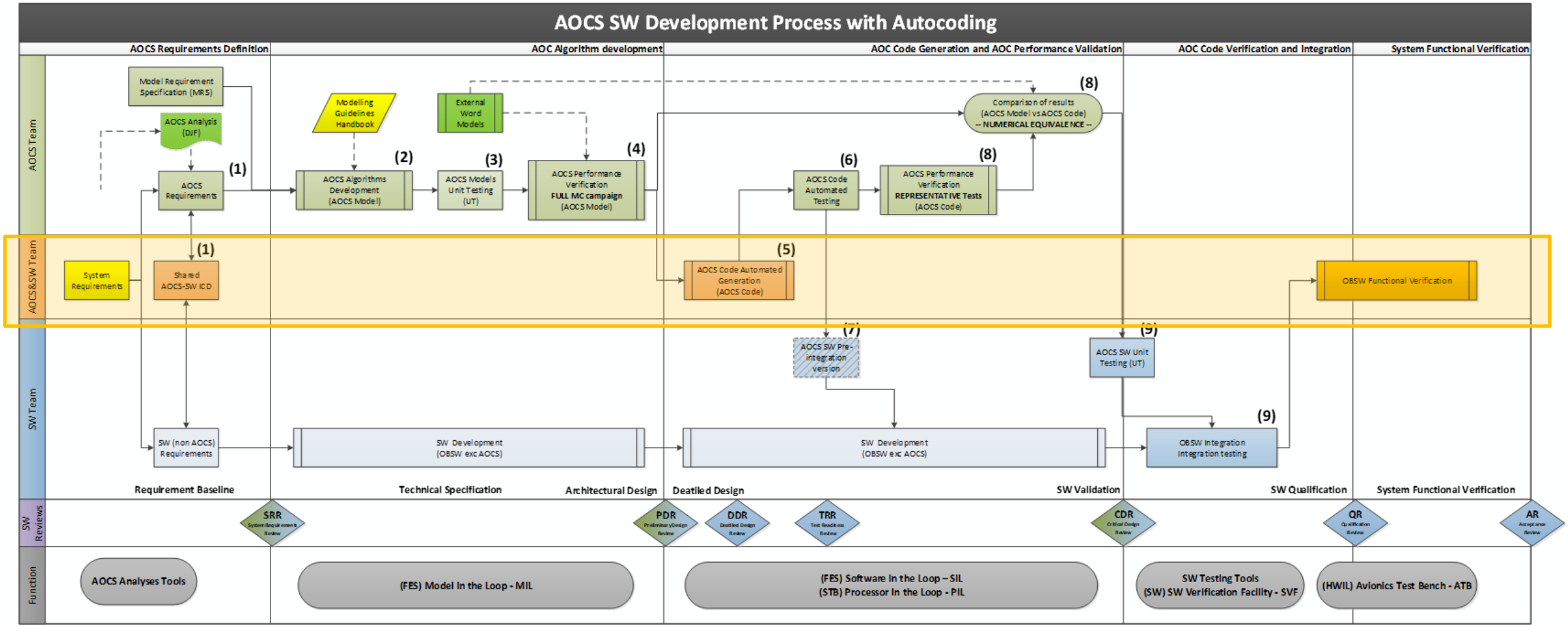
Autocoding process overview



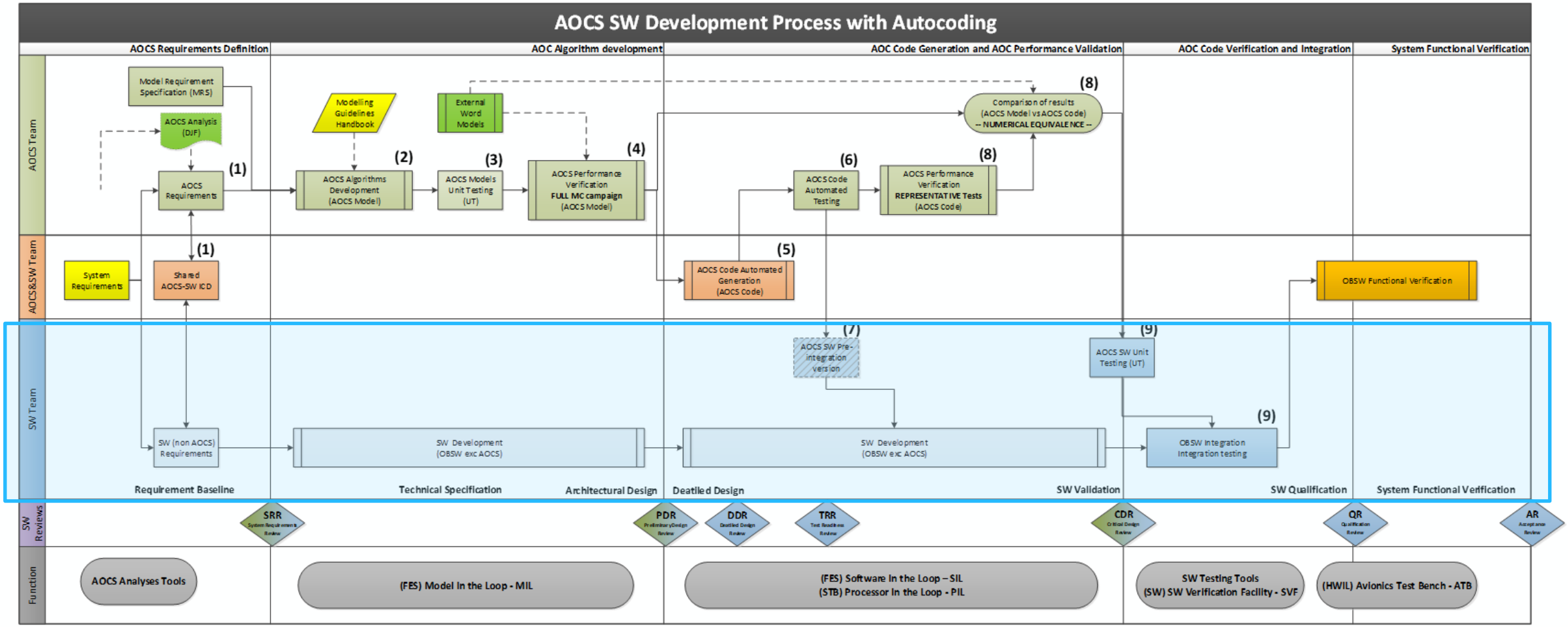
Development tasks – AOCS



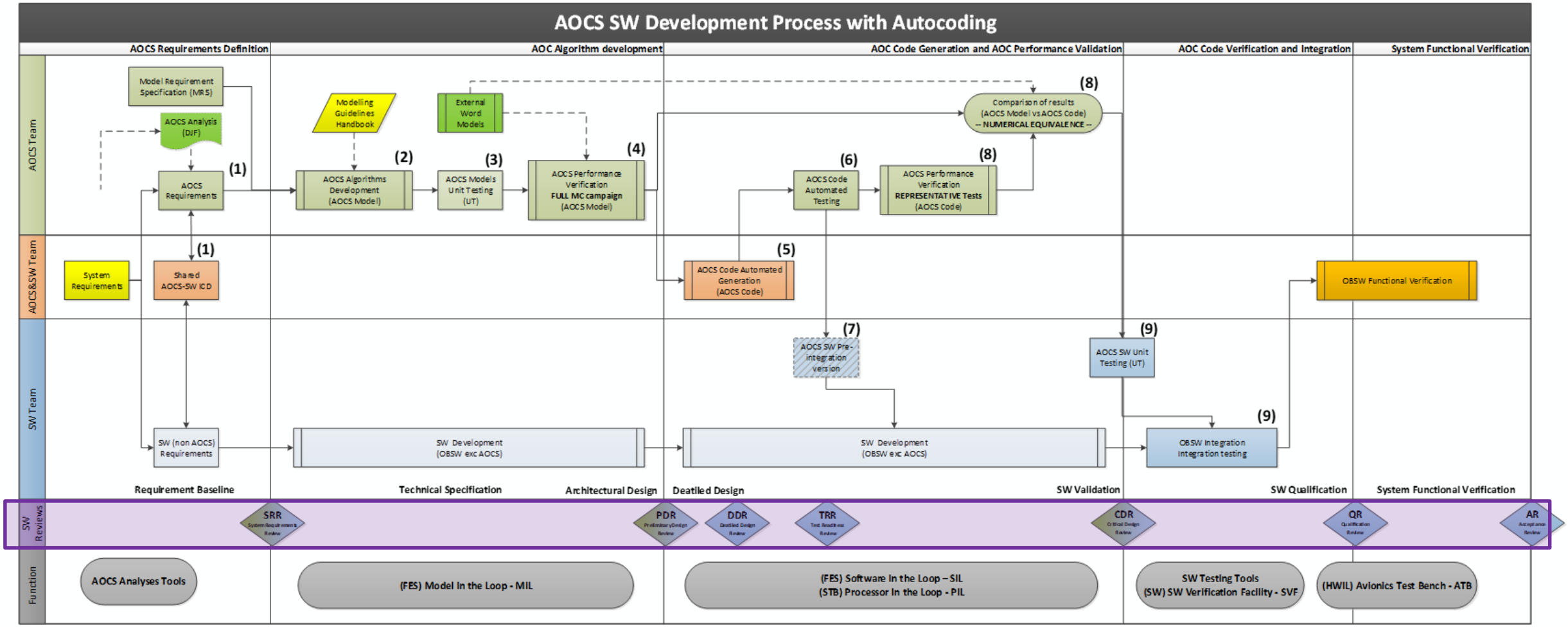
Development tasks - AOCS & SW



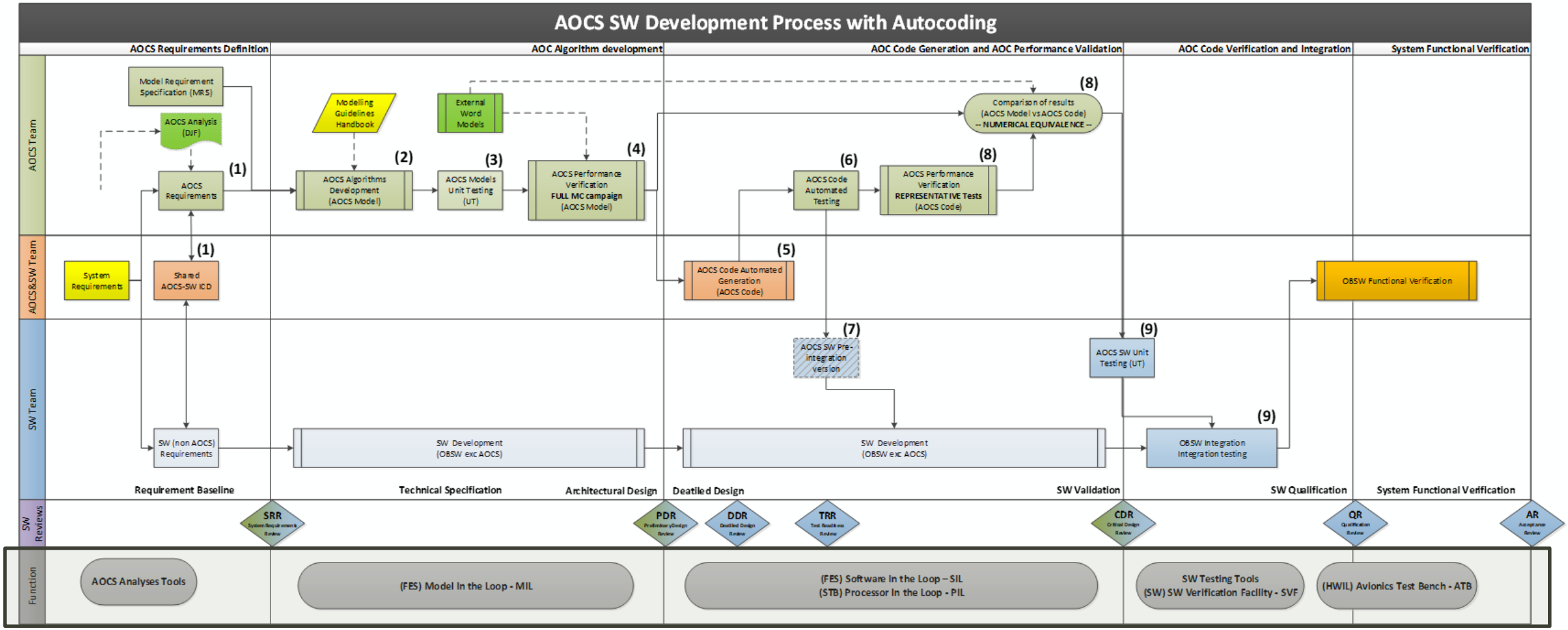
Development tasks - SW



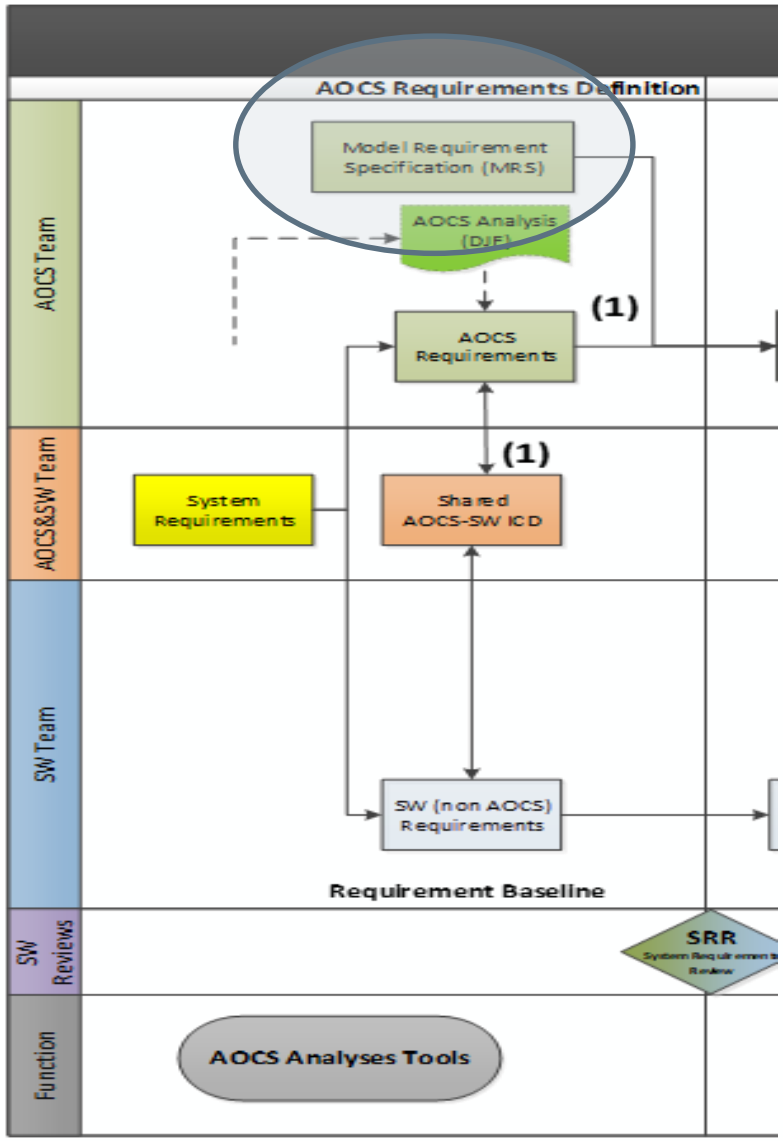
System Reviews timeline



Verification facilities



Requirements definition



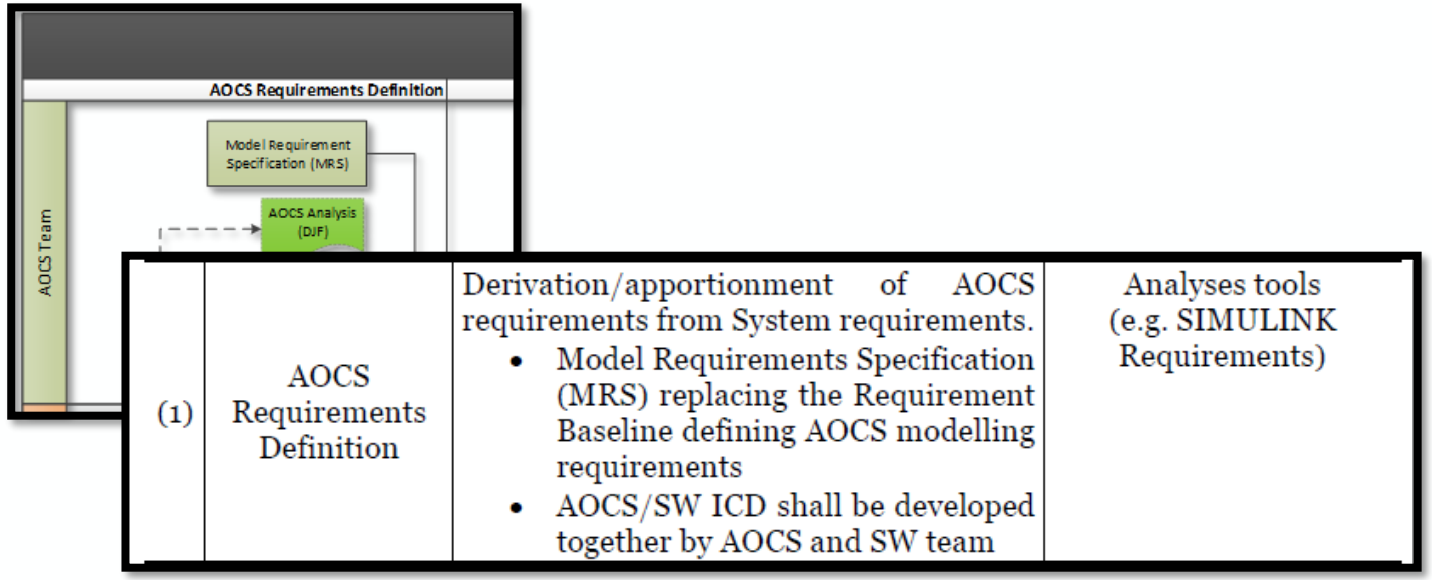
A. From system level to AOCS Subsystem:

- *Derivation of requirements: functional, performance, FDIR*
- *Justification of AOCS design architecture*

B. From system level to SW Subsystem:

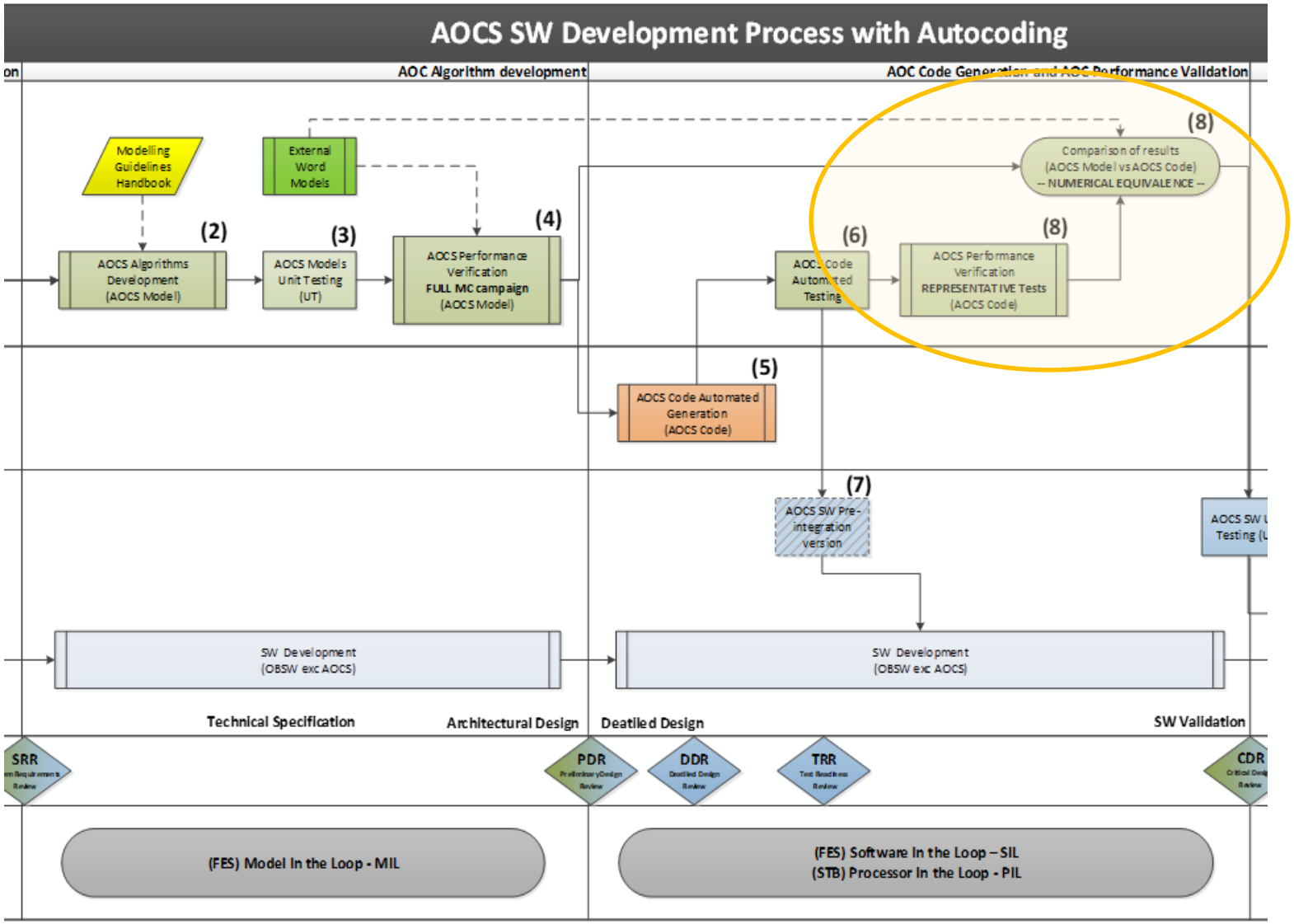
- *Derivation of SW requirements (non AOCS)*
- *Definition of AOCS SW Interface Requirements Document*

→ AOCS team shall start modelling following AOCS SW IRD



(1)	AOCS Requirements Definition	Derivation/apportionment of AOCS requirements from System requirements. • Model Requirements Specification (MRS) replacing the Requirement Baseline defining AOCS modelling requirements • AOCS/SW ICD shall be developed together by AOCS and SW team	Analyses tools (e.g. SIMULINK Requirements)

AOCS Performance Verification - Option1: MIL+PoE



A. AOCS performance verification MIL

- *AOCS model developed following Autocoding HB guidelines*
- *AOCS performance (pointing, stability, sensitivity..) verified on MIL wrt AOCS requirements baseline*
- *Justification of AOCS design*

B. Code generation:

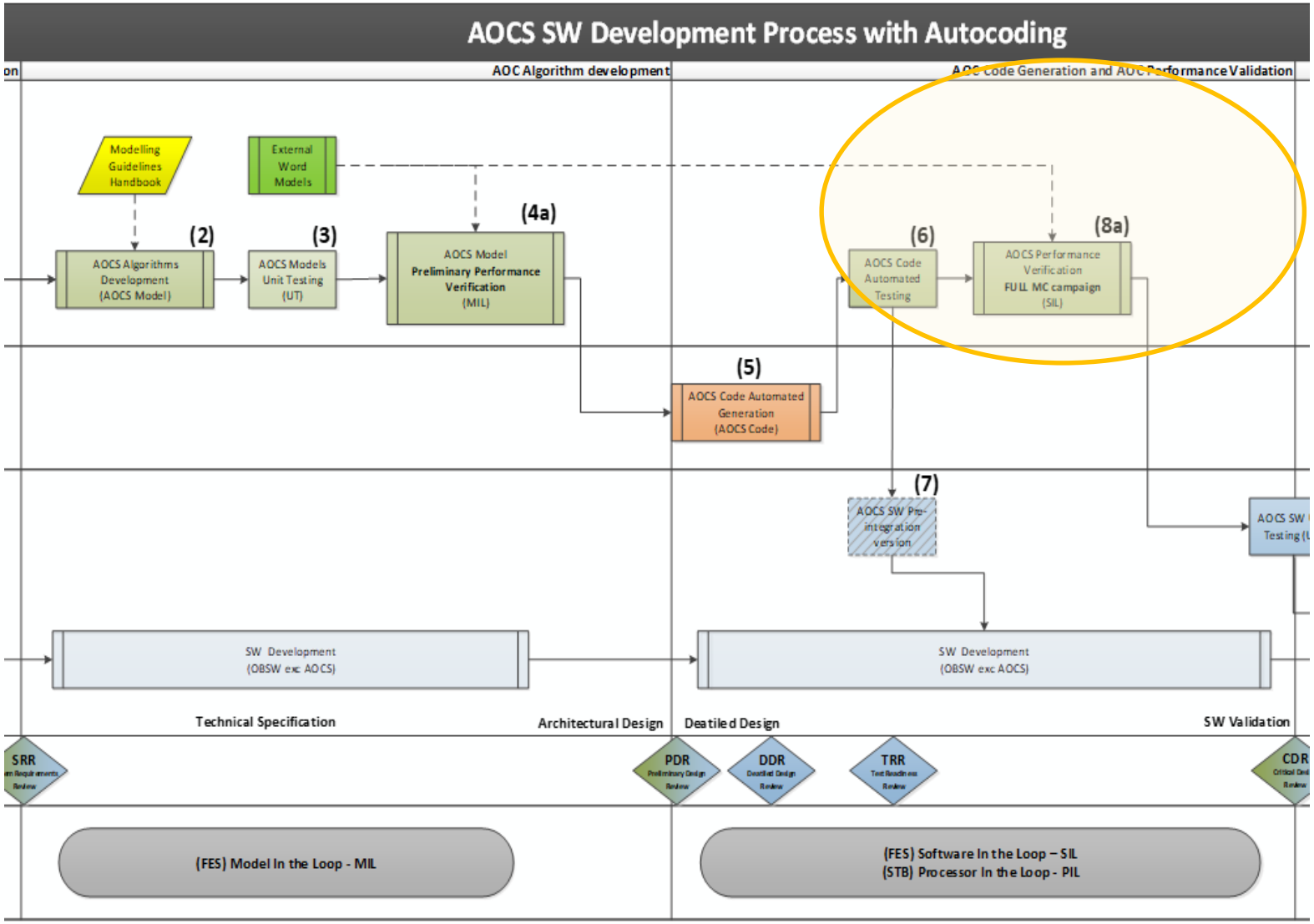
- *Code generation process configured as per HB*
- *Definition of AOCS Performance verification reference test cases (subset from MC campaign)*
- *Comparison of results MIL vs SIL → equivalence expected at last digital bit*

→ *The Proof of Equivalence (PoE) is mandatory to confirm performance verification on Model covers the verification of Code*

→ *The PoE can be achieved with accurate modelling and use of mathematical library*

→ *If PoE at last digit is not achieved, option 2 (see next slide shall be used)*

AOCS Performance Verification – Option2: SIL



A. AOCS development MIL

- *AOCS model developed following Autocoding HB guidelines*
- *Justification of AOCS design*

B. Code generation:

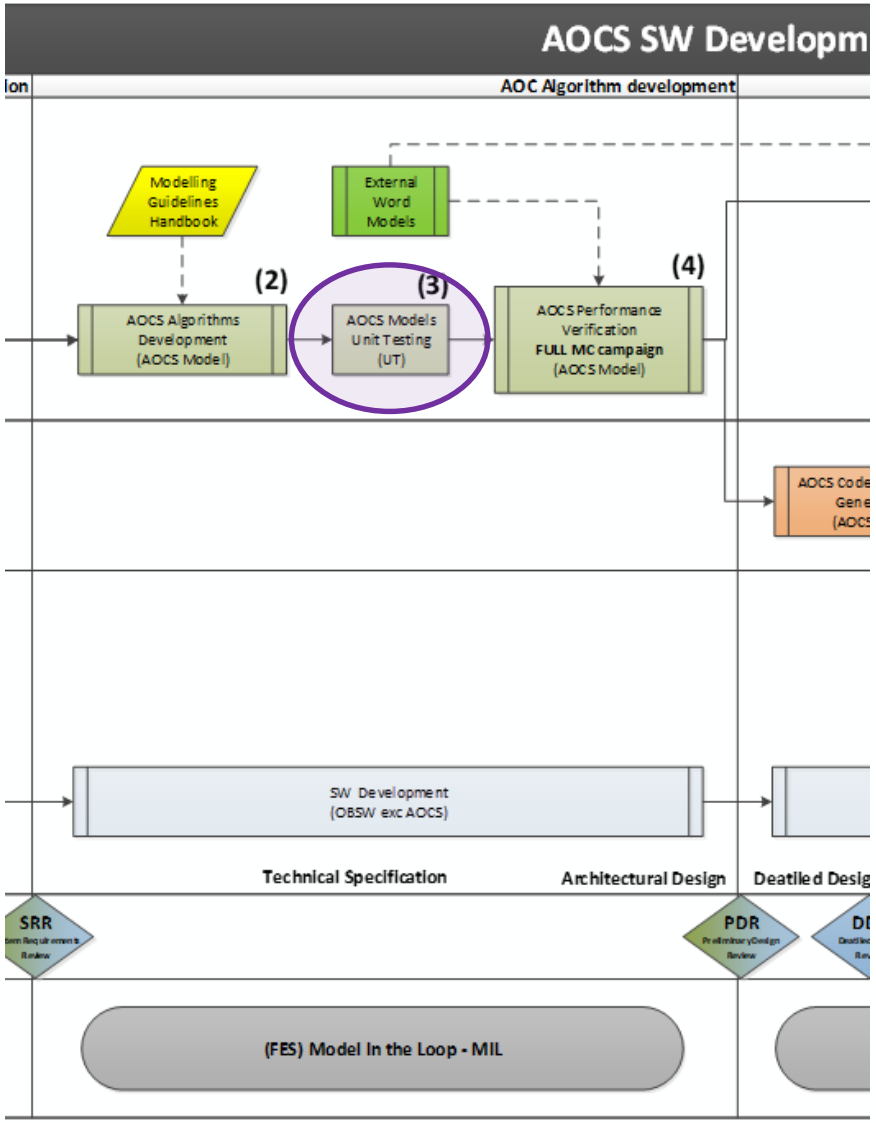
- *Code generation process configured as per HB*

C. AOCS Performance verification SIL:

- *Verification of AOCS performance (pointing, stability, etc..) running full MC campaign*

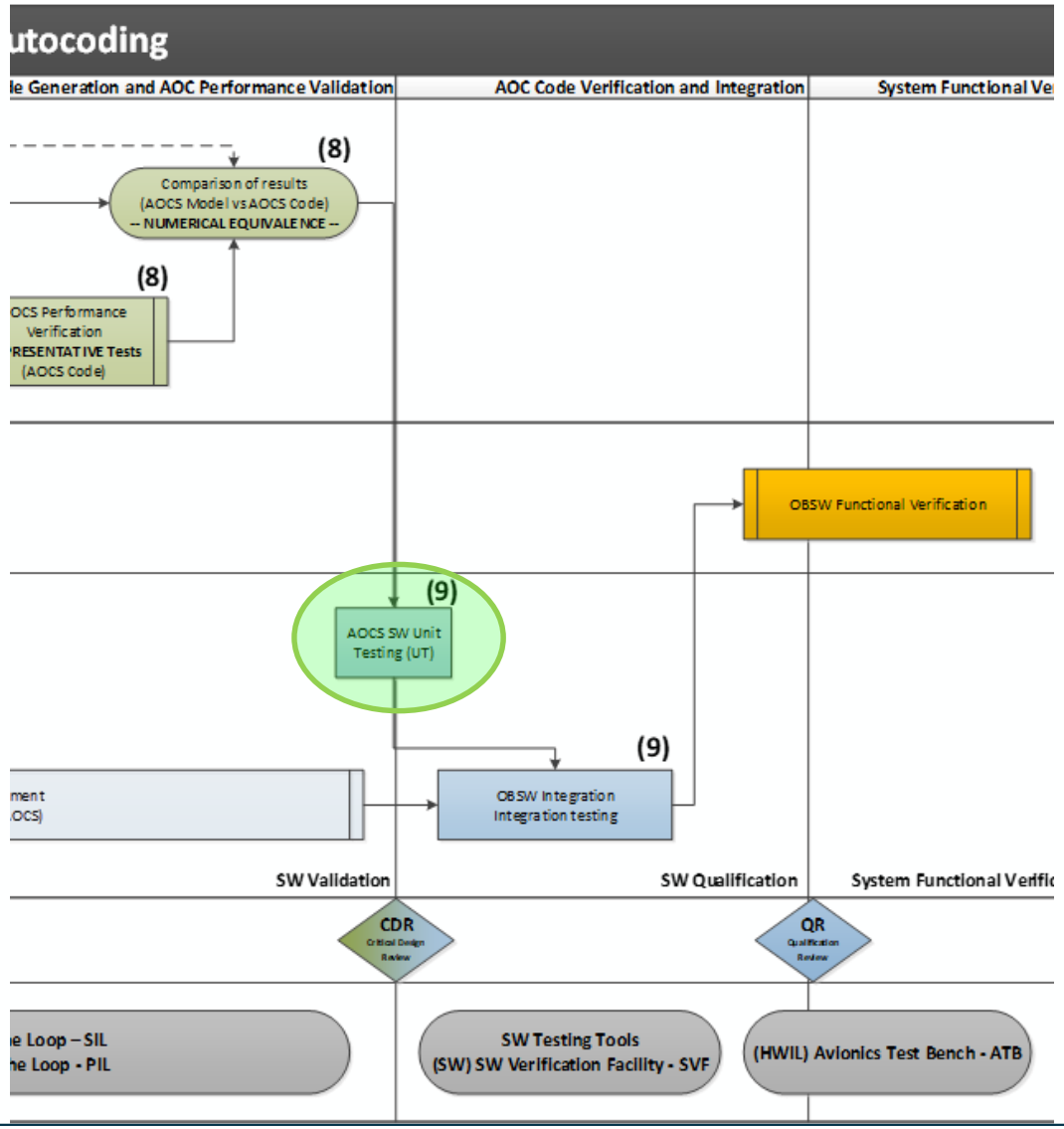
→ *The Code (final product) is used to verify AOCS Performance, no need of equivalence vs Model*
→ *Any iteration shall not allow Code modifications (changes through Model and Code generation)*
→ *Synchronization with reviews to be considered*

Model Unit Testing



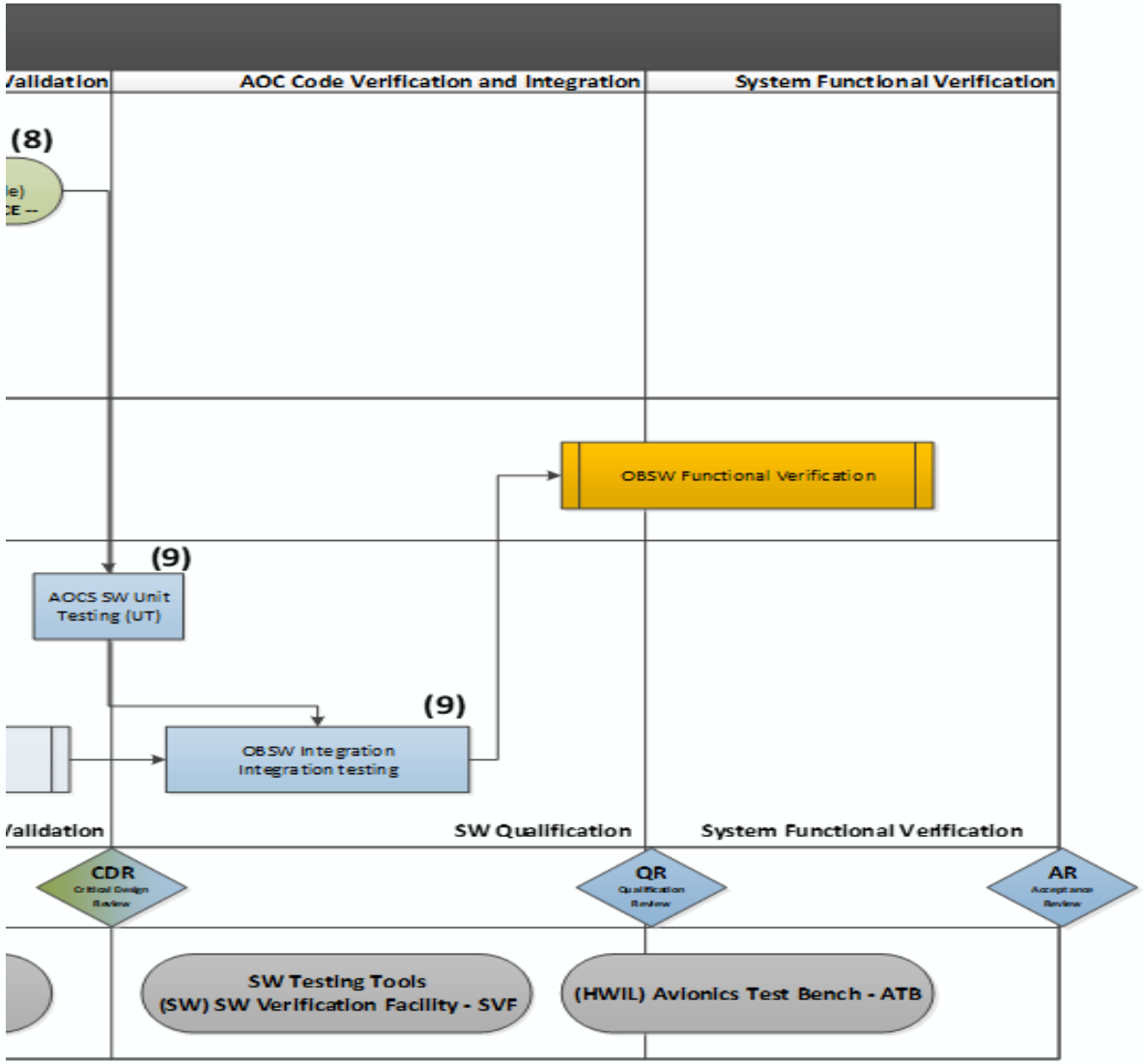
(3)	AOCS Models Unit Tests (AOCS Models)	The developed algorithms are subjected to Unit Testing where they are checked against modelling standard guidelines (generation of C-code) and open loop tests performed to be used as reference cases. Coverage tests are also performed to ensure sufficient model coverage with selected test harness. It needs to be reminded that model coverage does not imply coverage of the generated code. Note: the AOCS Models are used as Technical Specification (TS) for the automatic SW code generation, in the sense that SW behaviour will be validated against the Model behaviour	Modelling tools (e.g. MATLAB, SIMULINK)

SW Unit Testing



(9)	AOCS Code Verification and Integration (AOCS SW)	The generated and verified Code is delivered to the SW team to undergo the SW Unit Tests and SW performance tests as per standard SW development plan and the integrated in the On Board SW (OBSW) for further qualification and acceptance testing before final delivery to system for functional verification (as per standard process). It is noted that unit test a SW level have a different scope than unit test at model level (3). The UT at model level are intended to address modelling aspect while the one at SW level are intended to address code coverage and robustness approach as per E40C 5.5.3.2. Between the two set of UT there is a certain level of overlap and the scope of the SW UT can be reduced if it can be shown that certain aspects are addressed in the model UT. It is also noted that current process allocate the unit testing at SW level to the SW team, but depending on team expertise and tool chain it could also be conceivable to allocate this activity to the AOCS team.	SW Testing Environment (SVF, PIL, HWIL)

Delivery of the generated code

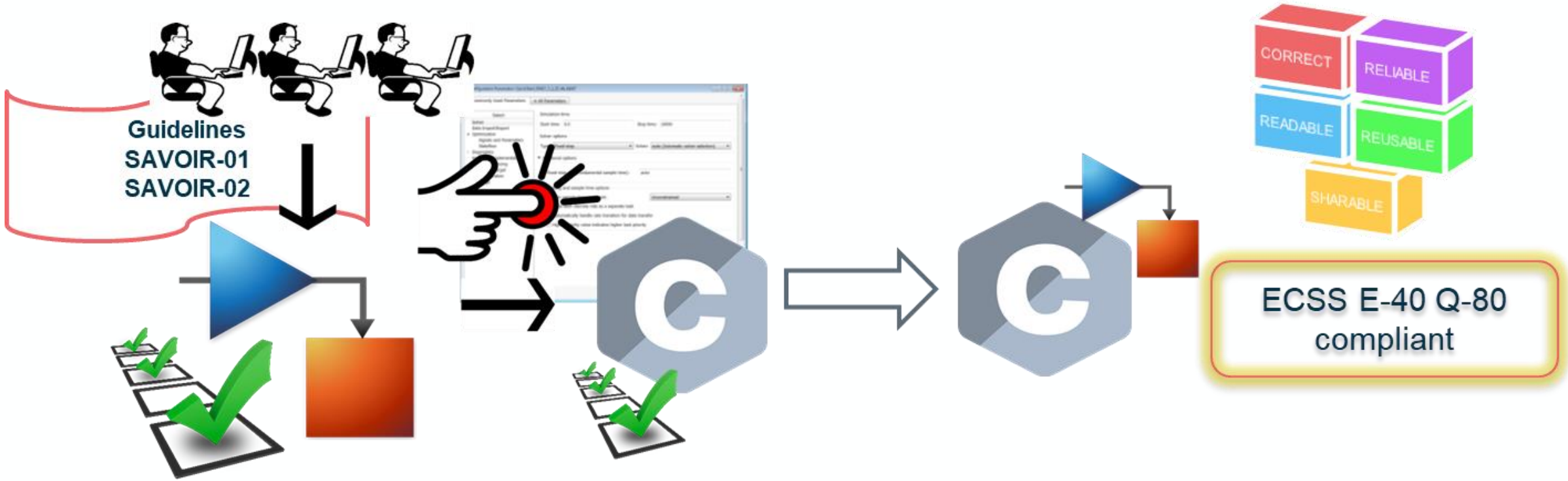


(9)	AOCS Code Verification and Integration (AOCS SW)	The generated and verified Code is delivered to the SW team to undergo the SW Unit Tests and SW performance tests as per standard SW development plan and the integrated in the On Board SW (OBSW) for further qualification and acceptance testing before final delivery to system for functional verification (as per standard process). It is noted that unit test a SW level have a different scope than unit test at model level (3). The UT at model level are intended to address modelling aspect while the one at SW level are intended to address code coverage and robustness approach as per E40C 5.5.3.2. Between the two set of UT there is a certain level of overlap and the scope of the SW UT can be reduced if it can be shown that certain aspects are addressed in the model UT. It is also noted that current process allocate the unit testing at SW level to the SW team, but depending on team expertise and tool chain it could also be conceivable to allocate this activity to the AOCS team.	SW Testing Environment (SVF, PIL, HWIL)

Outline of the presentation

- Background – Autocoding Working Group
- Vol I - Process: Autocoding generation process
- **Vol. II - Technology: Modelling Guidelines**
- Conclusions

Technology: modelling guidelines, code generation



Modelling Guidelines objectives

A. Modelling guidelines scope

- *Review the guidelines category, mandatory only when no alternative exists*
- *Express guidelines in terms of objectives, without providing implementation details*
- *Identify guidelines linked to toolbox used and those independent*

C. Coder Configuration settings

- *The code configuration settings shall be defined and applied at each generation to avoid differences*

D. Configuration control

- *The model versioning shall be kept in configuration control. Several methods exist to trace changes into the model (model block, each subsystem, etc..)*

Each guideline will be composed by the following fields:

Guideline ID:

- The ID is a unique reference for each guideline (between brackets reference to existing guideline from where it has been derived)

Guideline Title:

- The title is a short, but unique description of the guidelines area of application (e.g. '*length of names*').
- The title is used for the Prerequisites field and for custom checker-tools.
- The title text should appear with a hyperlink that links to the guideline.

Priority:

- Mandatory / Strongly recommended / Recommended

Description:

- This field contains a detailed description of the guideline.
- If needed, images and tables are included.

Rationale:

- Reasons the rule should be applied
- Mapping to improved quality index (e.g. readability, ..)
- Consequence/disadvantage in case the guideline is not applied

AOCS Modelling Guidelines

ID	SAVOIR-SE-050 (ID_ref:AUTO-MCR-070)
Title	Model Reference Library Blocks
Priority	Mandatory
Description	Model reference blocks shall be used at the unit test boundaries of model components with explicitly specified interfaces for data type and dimension. Model reference with explicitly specified interfaces should also be used whenever a component (and its generated code) should be functionally independent of the parent model.
Rationale	Changes made to the SIMULINK model within the model reference are guaranteed not to functionally change the generated code outside of the model reference. Changes made to the SIMULINK model outside of the model reference are guaranteed not to functionally change the generated code inside of the model reference. This guarantee cannot be asserted for other modelling components (including library blocks), because of the way Embedded Coder attempts to optimize generated code within a model.

Modelling guidelines classification

Modelling guidelines have the objective to ensure generated code is:

CORRECT — MANDATORY

- The code generation process shall work properly and free from errors, generated code shall correspond to the model.

RELIABLE — STRONGLY RECOMMENDED

- The generated code shall be fully equivalent to the model to ensure verification is valid

READABLE, REUSABLE, SHARABLE — RECOMMENDED

- Despite no manual changes to the generated code are foreseen, the readability shall be kept for code inspection, debugging, etc...
- The possibility to reuse or share generated code (industrial organizations)shall be foreseen by code readability

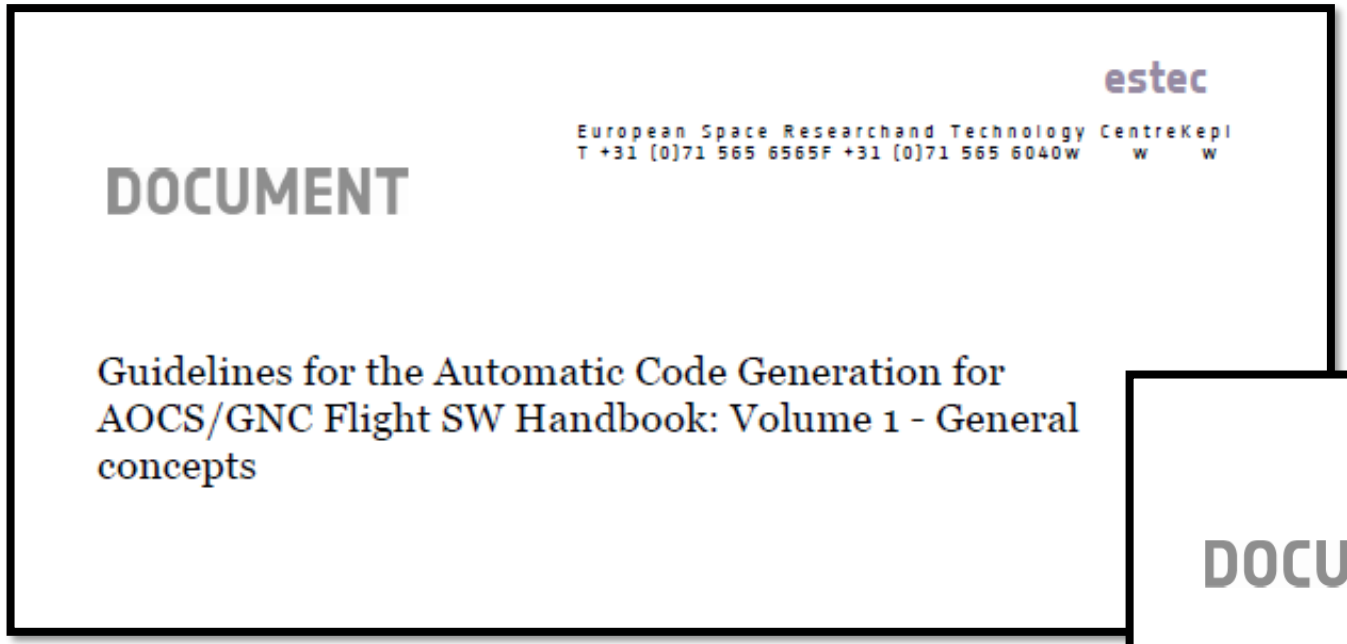
MANTEINABILITY — RECOMMENDED

- The maintenance of auto-generated code is meant to be achieved without manual intervention, but acting on the model and re-generating the code. This possibility shall be maximized by modular architecture and traceability

Outline of the presentation

- Background – Autocoding Working Group
- Vol I - Process: Autocoding generation process
- Vol. II - Technology: Modelling Guidelines
- **Conclusions**

SAVOIR Autocoding handbook SAVOIR-HB-005_i1

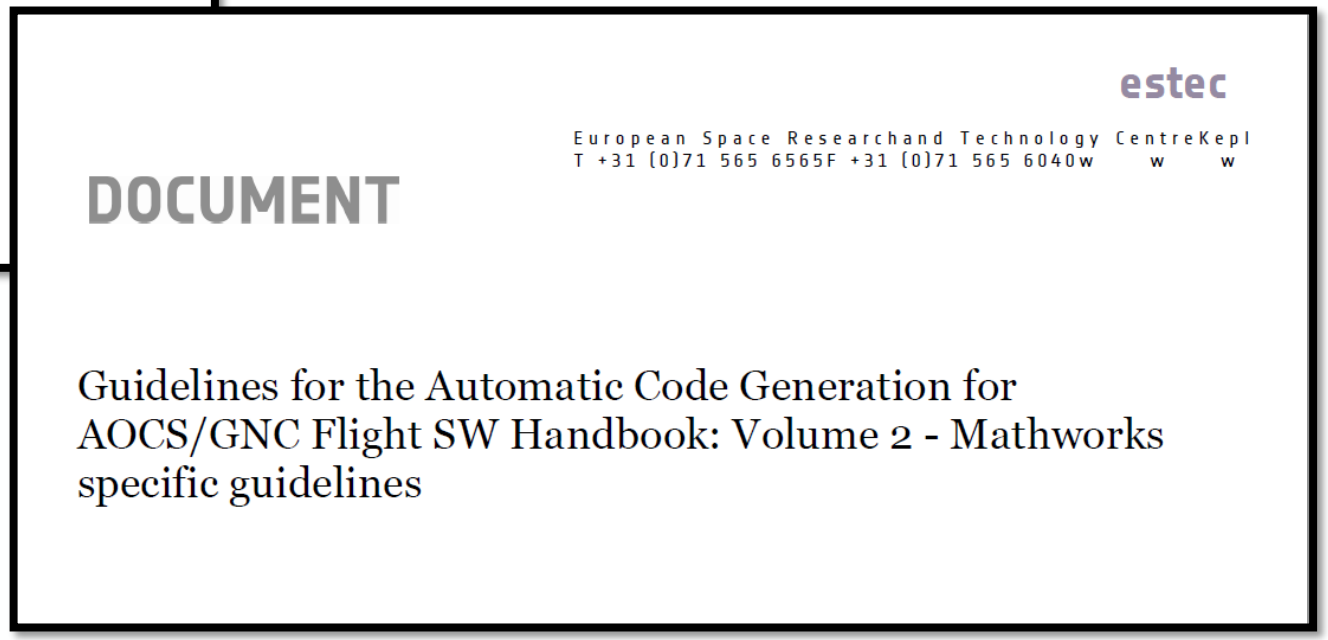

estec

European Space Researchand Technology CentreKepl
T +31 (0)71 565 6565F +31 (0)71 565 6040w w w

DOCUMENT

Guidelines for the Automatic Code Generation for AOCS/GNC Flight SW Handbook: Volume 1 - General concepts

The handbook is available in the ESA Space Software Repository
https://essr.esa.int


estec

European Space Researchand Technology CentreKepl
T +31 (0)71 565 6565F +31 (0)71 565 6040w w w

DOCUMENT

Guidelines for the Automatic Code Generation for AOCS/GNC Flight SW Handbook: Volume 2 - Mathworks specific guidelines


Future European Projects
SAVOIR
Verom


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
dziękuję
bedankt
спасибо
Баярлалаа
рахмат
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
terima kasih
arigatô
dakujem
grazie
dank je
tapadh leat
mauruuru
hvala
takk
мерси
감사합니다
ありがとう
ευχαριστώ
sagolun
blagodaram
faafetai lava
kiitos
dankie
enkosi
хвала
asante
manana
misaotra
malondo
paldies
mahalo
tack
welalin
barka
mersi
vinaka
dhanyavad
köszönöm
chnorakaloutioun
gracies
sulpáy
djiere dieuf
diolch
dhanyavadagalu
shukriya
trugarez
tanemirt
rahmet
xiexie
najis tuke
sobodi
dekuji
mesi
didi madloba
rahmat


IN MEMORIAM - FRANCESCO
GRASSINI