ThalesAlenia Space
a Thales / Leonardo company

SWPA ESA WORKSHOP
28th Sep 2023

A MODEL BASED APPROACH TO SW QUALITY MODELS

FROM MATLAB/SIMULINK MODEL TO QUALITY METRICS , AN AOCS EXPERIENCE

Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

ThalesAlenia Space
a Thales / Leonardo company

TABLE OF CONTENTS

THE TEAM

SW QUALITY

- SW PA responsibile

Ester Maio (TASinI , Rome)

- SW PA supervisor

Gianluca Caruso (TASinI , L'Aquila)

SW ENGINEERING

- SW ENG responsible

Alberto Petrucci (TASinI, L'Aquila)

- GSSI support

Patrizio Pelliccione (Computer Science Department at GSSI)

- TAS-I Attitude and Orbit Control System(s) (AOCS) responsibles

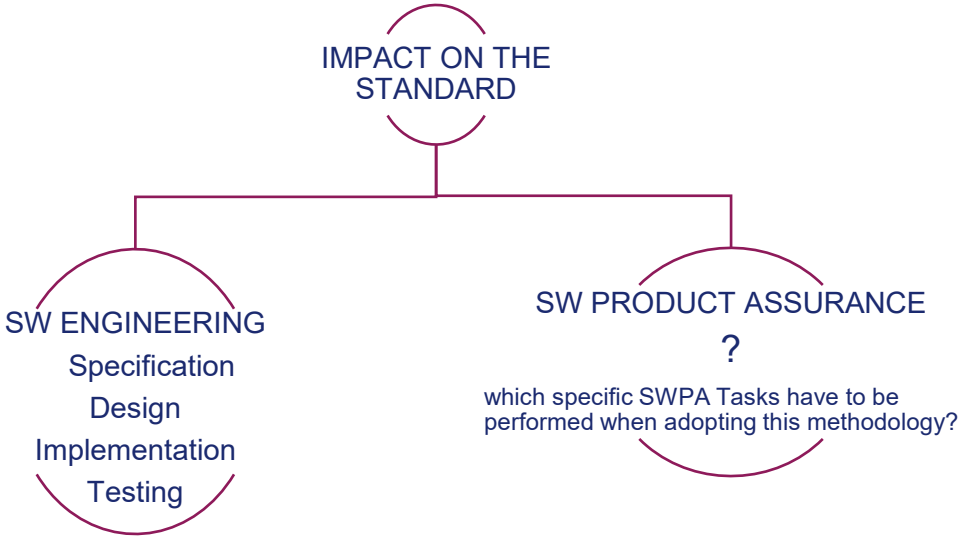
Marco Anania, Anio Beqiri (TASinI, Rome)

Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

INTRODUCTION

/// The model-based representation of the requirements or the design of a system has a lot of advantages. It is possible to directly generate the code associated to the modeled system automatically; that means :

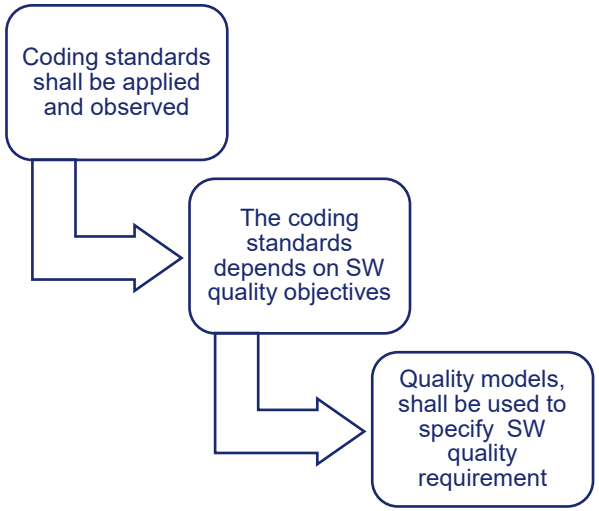
- Faster development and validation
- Earlier and easier verification and validation by means of simulation



Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

INTRODUCTION

/// In ECSS-Q-ST-80C rev1 there is a dedicated section for the automatic code generation, in particular the clause 6.3.4 highlights the importance of the verification of the coding standards implementation through the application of specific software quality objectives



Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

THE PROBLEM

/// The characteristics used to specify the quality models (basis for the identification of process metrics) are the same for classic code development and automatic code development. This point raises important questions

- Are all these classic metrics suitable for assessing also the quality of the automatically generated code?
  - which of the existing classic code metrics do already cope with such quality evaluation
- Is it necessary to define new quality properties that ensure the expected quality of the generated code? Can this properties be independent from the adopted language / tool?
- Is it necessary to define some models quality metrics that combined with the code metrics for the quality evaluation?

Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

SW QUALITY MODEL PROPOSAL

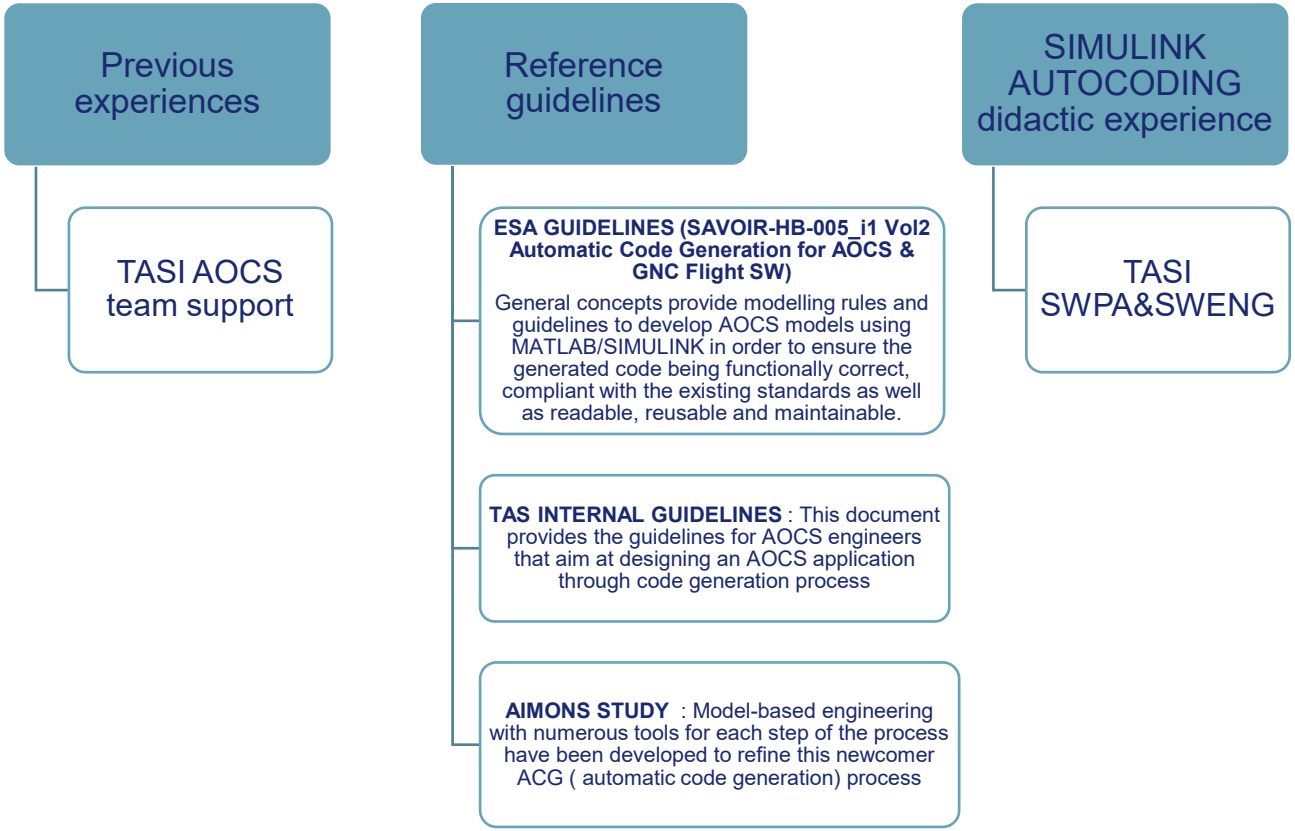
/// Our goal is to present a tailoring of a SW Quality Model fitting with the peculiarities of the automatic code generation

/// The ECSS-Q-HB-80-04 Handbook describes the approach to be taken for the definition and implementation of an effective and efficient metrication programme for the development of software in a space project

/// The tailoring of a metrication programme for an individual project requires several steps

- Identification of relevant project characteristics
- Selection of related base/derived metrics
- Allocation of measurement activities to life cycle processes
- Definition of evaluation / analysis criteria:
  - to use the metrication results as a means of monitoring and steering the project it is necessary to define limits or thresholds that individual metrics should not exceed
- Detailed definition of data to be captured

SW QUALITY MODEL PROPOSAL :HOW?



Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

SIMULINK AUTOCODING DIDACTIC EXPERIENCE :DESCRIPTION

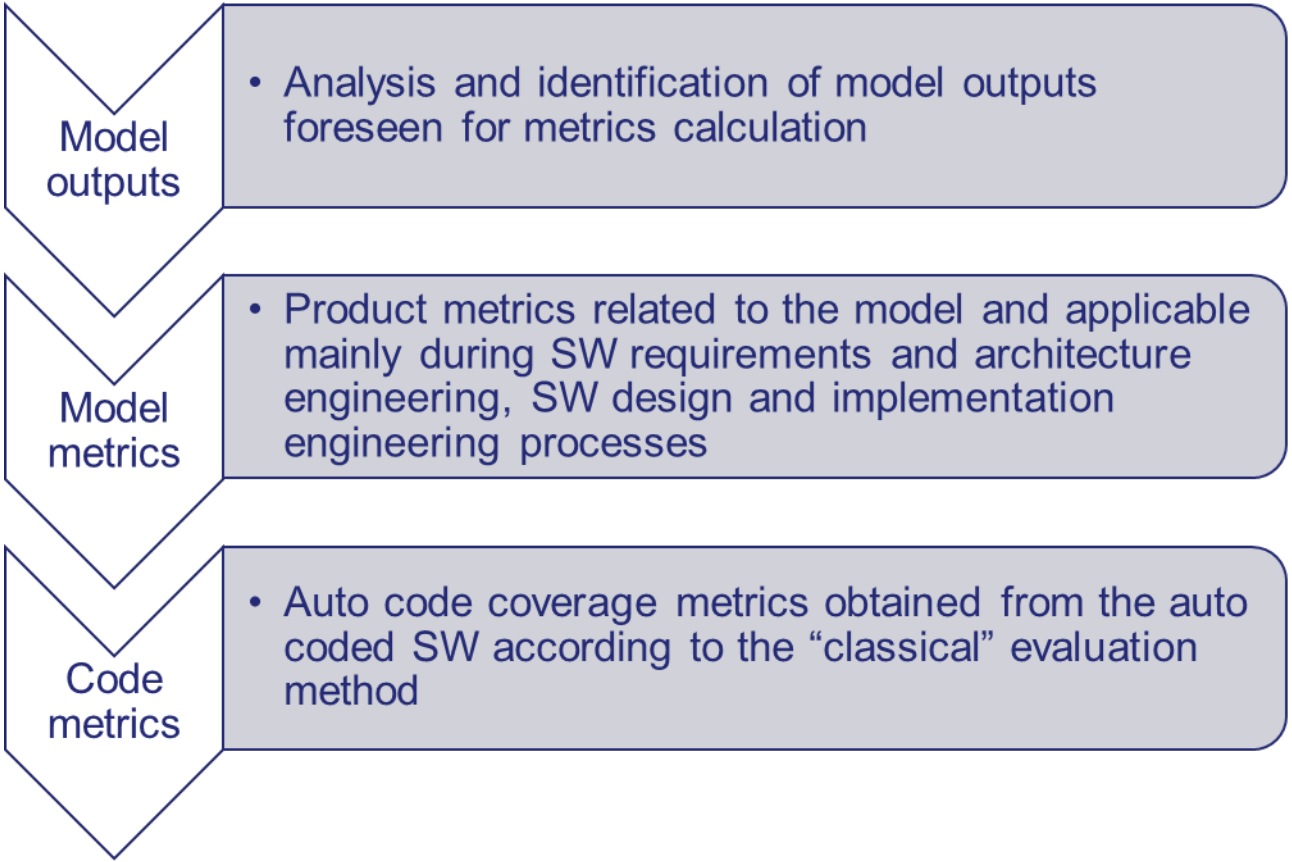
/// We used a free didactic on line model composed by three Sub-System (SS):

- The Controller SS: implemented using a PID
- The Sensors SS: used to measure the needed information from the SpaceCraft model
- The Space Craft SS

/// We applied the auto coding process to a single subsystem and to the whole system using the Simulink Coder that generates and executes C and C++ code from Simulink model

Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

SIMULINK AUTOCODING DIDACTIC EXPERIENCE :PROCESS



Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

SIMULINK AUTOCODING DIDACTIC EXPERIENCE :MODEL OUTPUTS

/// Coverage Report

- A .html report can be read by everyone and does not require a Matlab license

/// Model Architecture in a readable format

- A .xsl file for the architecture mapping with the generated code

/// Requirements report

- It can be a *.docx*, *.pdf*, *.htlm* and it can be a report for one or more sets of requirements

/// Model metric and complexity report

- It is a .html report that provides metrics about size, architecture, readability

/// Auto generated code

- For code and traceability metrics evaluation

Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

SIMULINK AUTOCODING DIDACTIC EXPERIENCE :MODEL METRICS

/// The Product metrics identified and applicable to the model are summarized in the table below:

(Main) characteristic	Sub characteristic	Metrics	First provided at	Frequency
PRODUCT RELATED CHARACTERISTICS (MODEL)				
Functionality	Completeness	Requirement allocation	PDR	Every Review
	Observability	Requirement implementation coverage (traceability to code)	CDR	Every Review
	Concistency/Efficency of model environment	Inling numeric values of block parameters	PDR	Every Review
	Correctness	Adherence to coding standards of Matlab function	CDR	Every Review
Maintainability	Portability, traceability	Model version/Code version	CDR	Every Review
	readability/traceability	Model Hierarchy/Complexity	PDR	Every Review
	Complexity	Cyclomatic complexity (VG)	PDR	Every Review
		Saturation on integer overflow	PDR	Every Review
		Maximum Subsystem Depth	PDR	Every Review

Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

SIMULINK AUTOCODING DIDACTIC EXPERIENCE :MODEL METRICS

/// For each metric we have identified

Main Characteristic
Sub Characteristic
Metric name
Goal
Owner / Producer
Target audience
Evaluation method
Formula
Interpretation of measured value
Life cycle phase
Applicability
Pre-conditions
Report format

Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

SIMULINK AUTOCODING DIDACTIC EXPERIENCE :MODEL METRICS EXAMPLE

A.7 Maximum Subsystem Depth

Main Characteristic	Maintainability
Sub Characteristic	*Complexity/readability*
Metric name	*Maximum Subsystem Depth*
Goal	this metric calculates the depth of subsystems in the model. This value should be limited to limit the depth of the functions in the generated code
Owner / Producer	Owner: development leader Producer: development team
Target audience	Development leader, SW PA manager
Evaluation method	model metrics and complexity report
Formula	Value in the report: subsystem depth for each component in the hierarchy
Interpretation of measured value	the value should be defined depending on project. A reference value can be 15
Life cycle phase	Collected during SW design and implementation engineering processes. Provided at CDR and updated afterwards as required.
Applicability	- MANDATORY
Pre-conditions	*collect data for this metric using the Model Advisor in Simulink*
Report format	Textual (see example below)
Other remarks	

Subsystem depth metric (06-Apr-2023 09:15:20)

Display depth of subsystems in the model or subsystem.

Total Subsystem Depth: NaN

Passed

Component	Subsystem Level	Subsystem Depth
PID_Controller	0	2
PID_Controller/SpaceCraft	1	1
PID_Controller/Trajectory	1	0
PID_Controller/Controller	1	0
PID_Controller/Sensors	1	0
PID_Controller/SpaceCraft/Inertia Matrix	2	0
PID_Controller/SpaceCraft/Gravity Vector	2	0
.../Centrifugal and Coriolis Matrix	2	0
PID_Controller/Subsystem	1	0

Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

FUTURE LINES OF WORK

/// Model metrics vs Code metrics analysis

- Comparison of model complexity metric and code metric . Are necessary different thresholds?

/// Application of traceability between in models and software requirements

/// The safety and security topics: security requirements and other critical requirements must be included in the model design development to be included in the generated code

/// SW tests (unit, integration and validation): the tests performed at model level have to be performed also at code level

/// Process related metrics

/// Testing on AOCS development

- Define new thresholds for auto generated code complexity

Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011

CONCLUSION

/// As specified in ECSS E 40: the trend in SW ENG is to move from a document centric development towards a model centric development and the interpretation of the standards has to evolve with the appearance of the model.

/// This paper wants to be a starting point common metrication program in order to evaluate SW auto generated code quality and a little step forward for the standard evolution.

/// A well-defined and thoughtful approach to Software Quality Assurance is necessary to allow auto coding to become a trusted solution.

Date: 28/09/2023
Ref:
Template: 83230347-DOC-TAS-EN-011