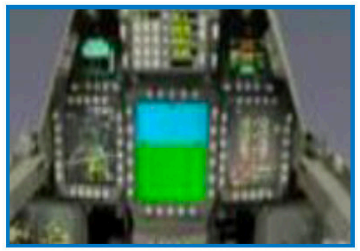
Approches basées modèles de l'ingénierie des exigences pour les systèmes embarqués critiques.

JFIE 2019

Amar Bouali
Inria / Université Côte d'Azur

Contexte

Développement de systèmes embarqués critiques, à dominante logicielle



Aerospace & Defense

500% increase in software lines of code (SLOC) in aerospace in 10 years

Automotive

100Mi software lines of code (SLOC) in modern vehicles

Railways

Ever increasing certification costs and project delays/costs overrun

Industrial Equipment

More than 380K software and system engineers work in the oil and gas industry

Energy & Nuclear

Software-based Instrumentation and Controls have become State of the art

Healthcare

Software Failures are Responsible for 24% of all Medical Device recalls

Pourquoi ce contexte ?

- Les développements logiciels des autres secteurs atteignent des complexités similaires
- Caractère critique en terme de sécurité, coûts d'ingénierie et d'impact économique et sociétal comparable

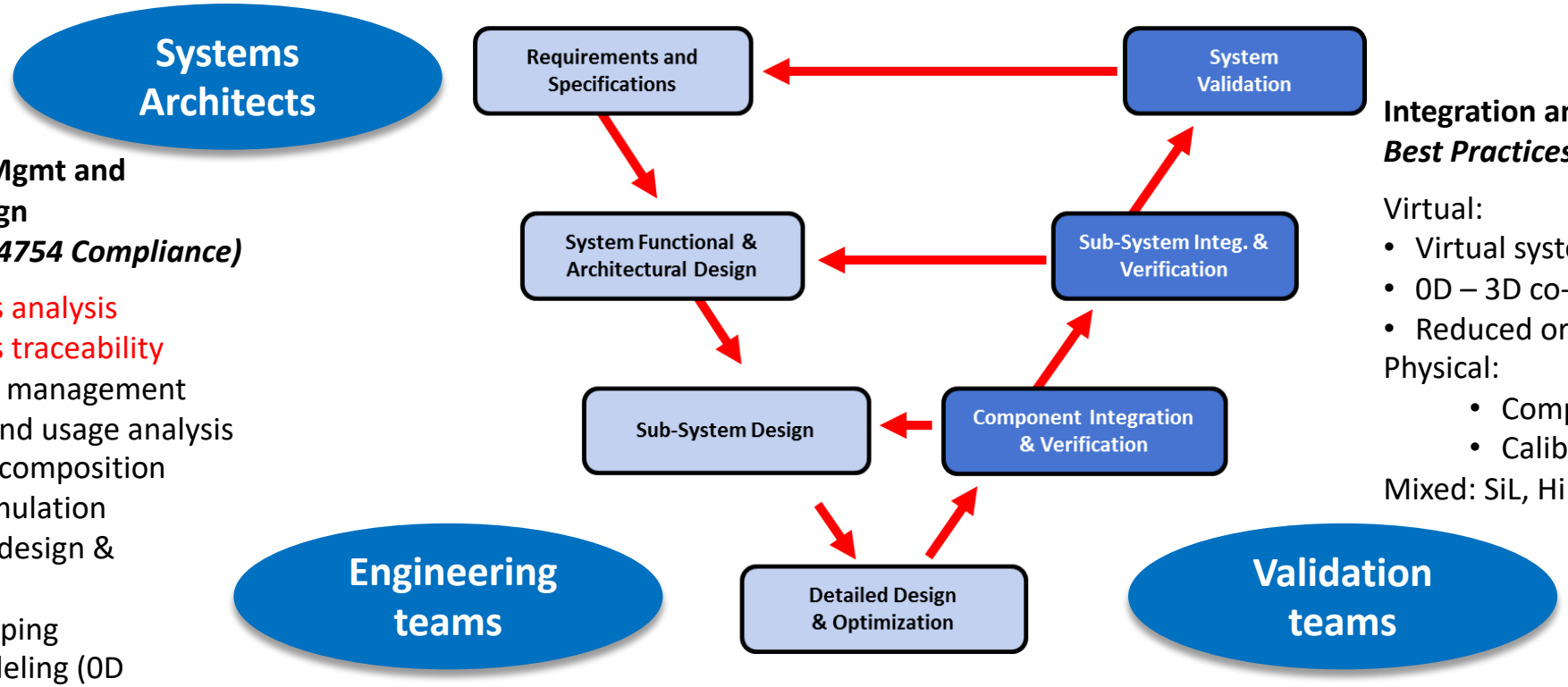
➔ Tirer les bonnes pratiques des domaines des systèmes critiques

- Ingénierie cadrée par des normes rigoureuses
- La qualité et la traçabilité des exigences au cœur de la certification
- Méthodes formelles permettant de maîtriser les efforts et coûts tout en améliorant la qualité.

Agenda

1. Défis de l'ingénierie des systèmes embarqués critiques
2. Ingénierie des exigences et approches basées modèles
3. La traçabilité dans une approche basée modèles
4. Exemple d'une solution industrielle
5. Conclusion

Défis de l’ingénierie des systèmes critiques



Requirements Mgmt and Functional Design
Practices (ARP 4754 Compliance)

- Requirements analysis
- Requirements traceability
- Configuration management
- Operational and usage analysis
- Functional decomposition
- Functional simulation
- Architectural design & selection
- Rapid prototyping
- Behavior modeling (0D simulation)

Integration and Validation
Best Practices

Virtual:
- Virtual system integration & simulation
- 0D – 3D co-simulation
- Reduced order modeling

Physical:
- Component Hardware testing
- Calibration

Mixed: SiL, HiL, TiL

Detailed Design and Optimization
Best Practices

Hardware Design
- CAD
- Single physics
- Multiphysics
- Optimization

Electronics Design
- ECAD
- EDA
- Circuit analysis
- 3D physics
- Multiphysics
- Optimization

Software Design (DO178B/C Compliance)
- Model-based controls design
- Model-based display design
- Automatic code generation and certification

Les défis

Assuring Functional Safety and Security
Managing Design Complexity
Optimizing Overall System Performance
CERTIFIED
Reducing Embedded Software Costs
Reducing Physical Validation Costs

Défis opérationnels

Manage Design Complexity

- Manage Requirements & Traceability
- Manage Functional and Architectural Designs
- Produce Interface Control Documents (ICDs)
- Comply with standards

Reduce Embedded Software Costs

- Cost of Embedded Code production
- Cost of Embedded Code testing
- Cost of Certification
- Comply with industry standards

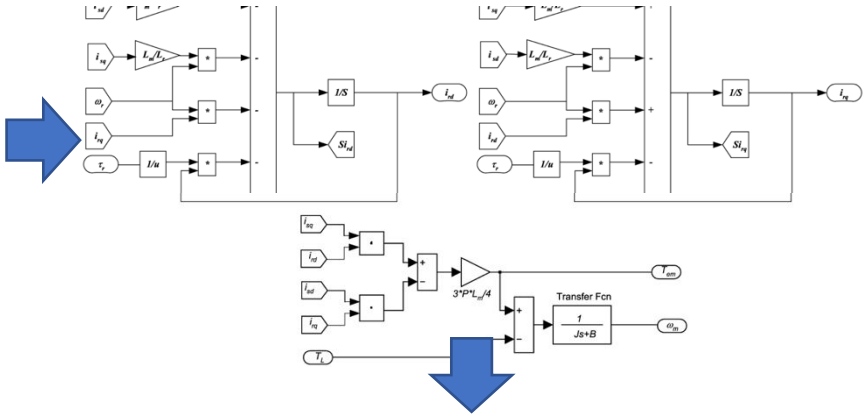
Optimize System Performance & Eliminate Integration Failures

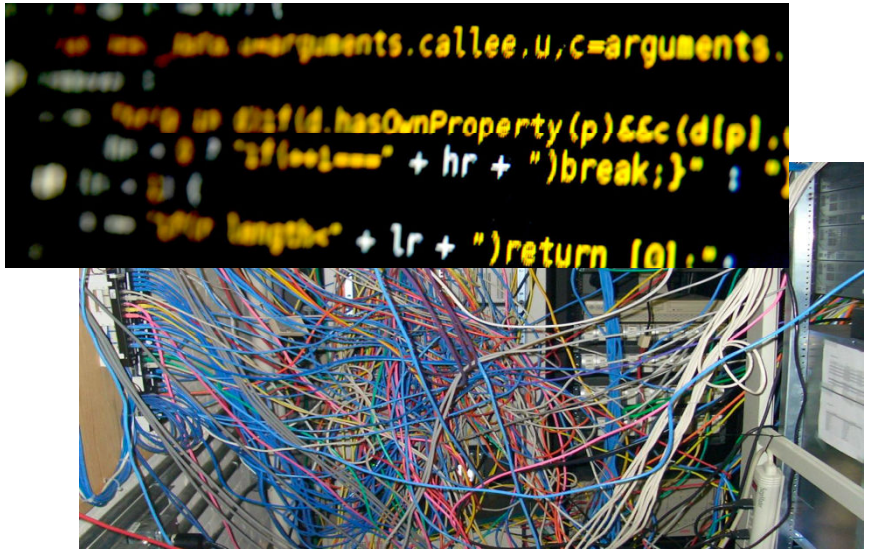
- Enable Virtual System Prototyping
- Optimize system performance
- Eliminate late stage integration failures
- Reduce physical and hardware-in-the loop testing

Limites des approches traditionnelles

Req_1: ths system shall emit an alarm…

Req_2: all doors shall be closed upon…





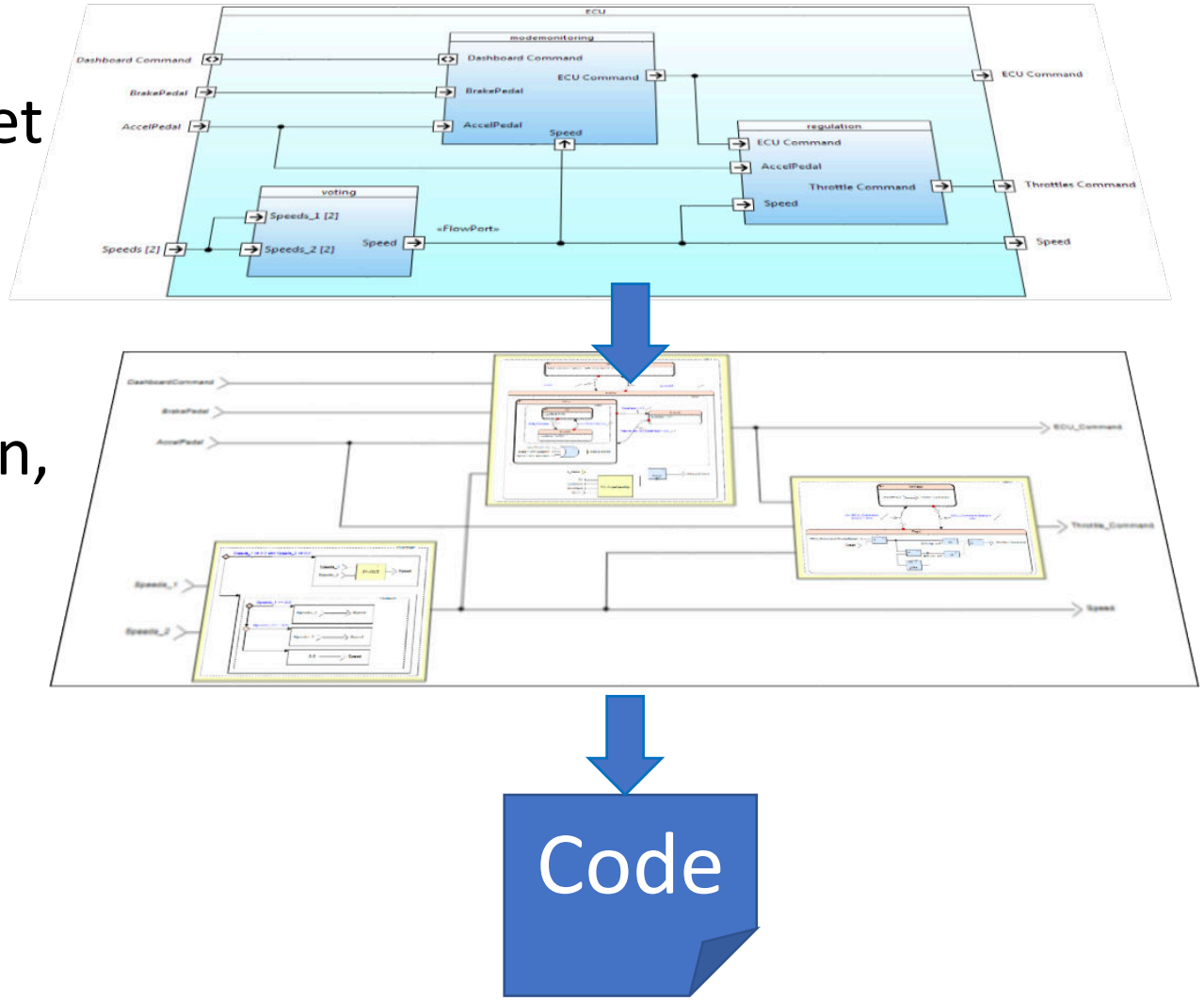
- Exigences incomplètes, ambiguës
- Prise en charge avec des approches métiers spécifiques
- Traçabilité et communication difficile entre les activités d'ingénierie
- Détections d'erreur tardives
- Passage à l'échelle difficile pour les systèmes modernes

Ingénierie des exigences et approche basée modèle

L’exemple de l’aéronautique

Principes du basé-modèles

- Notation et langages graphiques formels pour l'ingénierie système et logiciel
- Environnement logiciel pour la conception, l'analyse, la vérification, et la génération automatique de code
- Réduction des coûts jusqu'à 40%

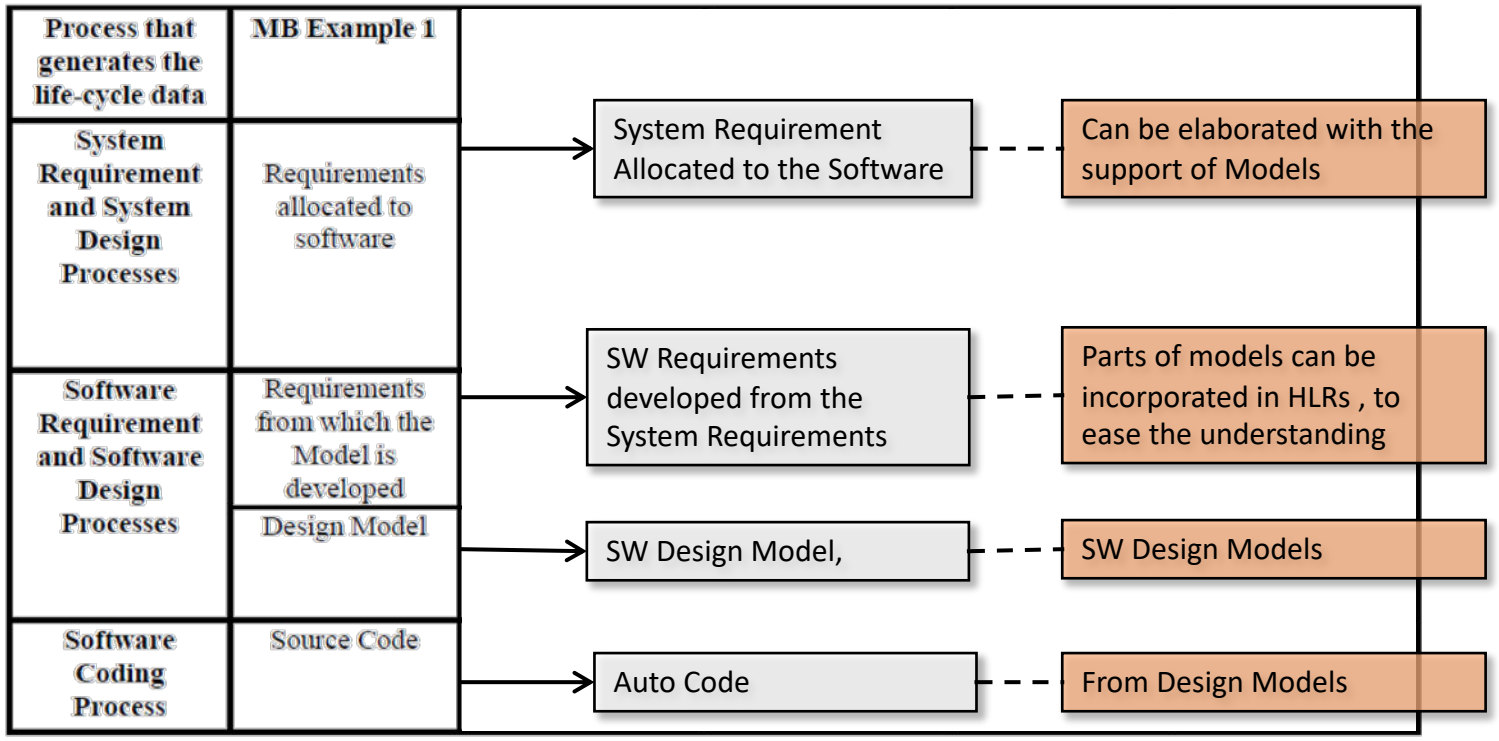


Point de vue de la Certification

- Aéronautique : approches formelles basées-modèles encouragées
  - ARP-4754A pour l'ingénierie système
  - DO-178C pour l'ingénierie logicielle
  - Supplément DO-331 : développement et vérification basée modèle
  - Supplément DO-330 : qualification des outils

- Autres industries
  - Ferroviaire: EN 50126-8, hautement recommandées
  - Automotive: ISO 26262, approuvée pour l'analyse de sûreté, défaillance, tolérance aux fautes et l'utilisation d'outils logiciels

L'approche basée modèles dans DO-331



- Un modèle est un moyen de représenter les exigences logicielles et d'architecture
- Un modèle doit être conçu complètement à partir d'exigences externes
- La simulation est un moyen d'effectuer la vérification de modèles

DO-330: Qualification d'outils

- Outils éliminant, réduisant ou automatisant une activité
- Niveau de qualification en fonction du risqué lié à l'outil
  - **Critère 1** : sortie de l'outil pouvant introduire une erreur dans le logiciel embarqué
  - **Critère 2** : automatisation d'une procédure de vérification pouvant échouer et justifiant l 'élimination ou la réduction de processus de vérification ou de développement lié au logiciel embarqué
  - **Critère 3** : un outil utilisé à ses propres fins pouvant échouer dans la détection d'erreur
- La traçabilité est au cœur des activités de qualification des outils

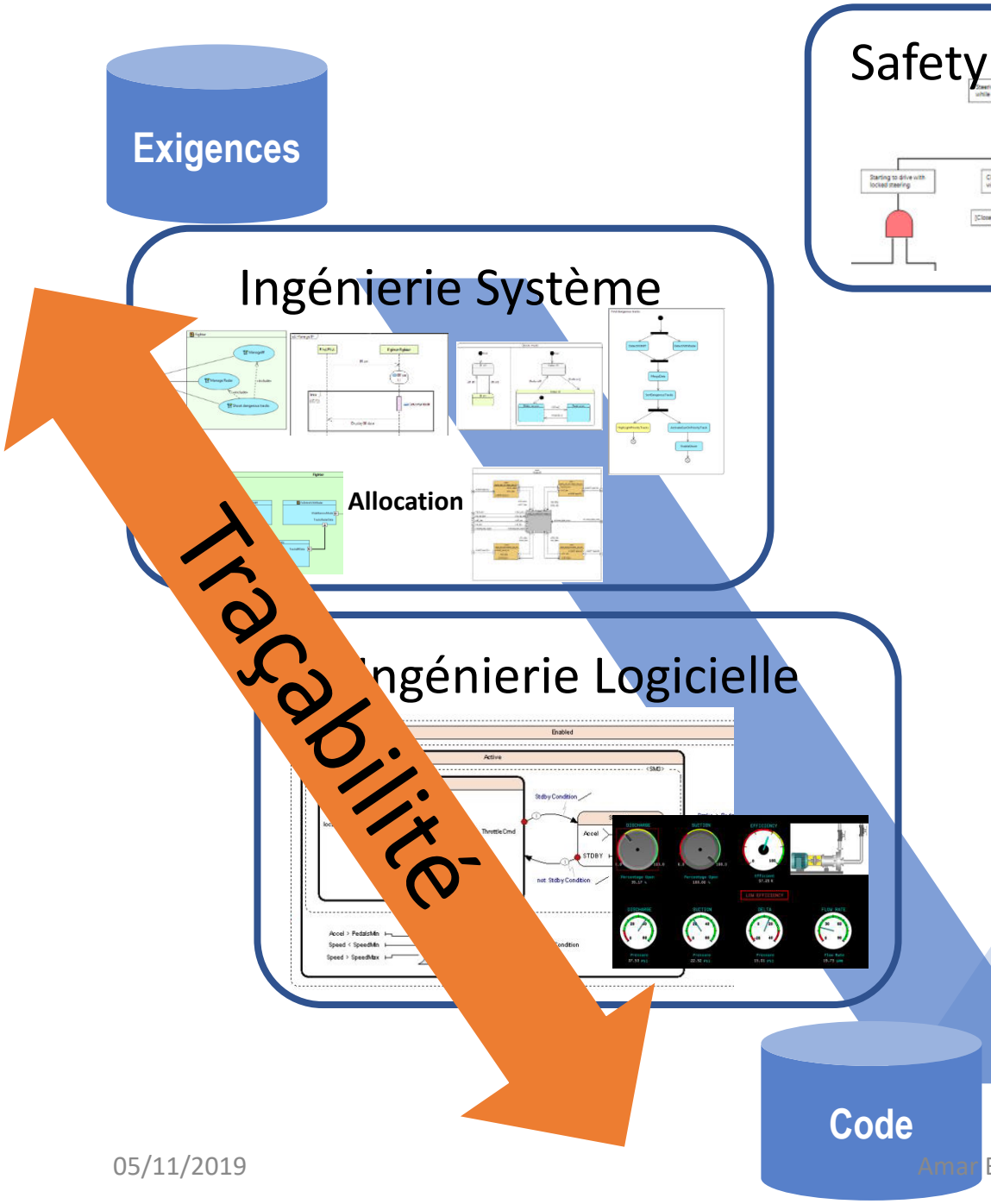
Générateur de code

Générateur de rapports de vérification

(DO-178C) Software Level	Criteria		
		2	3
A	TQL-1	TQL-4	TQL-5
B	TQL-2	TQL-4	TQL-5
C	TQL-3	TQL-5	TQL-5
D	TQL-4	TQL-5	TQL-5

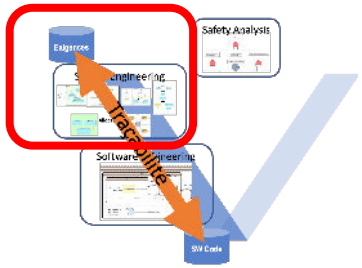
Level	Effect of anomalous behavior
A	Catastrophic failure condition for the aircraft (*e.g.*, aircraft crash).
B	Hazardous/severe failure condition for the aircraft (*e.g.*, several persons could be injured).
C	Major failure condition for the aircraft (*e.g.*, flight management system could be down, the pilot would have to do it manually).
D	Minor failure condition for the aircraft (*e.g.*, some pilot-ground communications could have to be done manually).
E	No effect on aircraft operation or pilot workload (*e.g.*, entertainment features may be down).

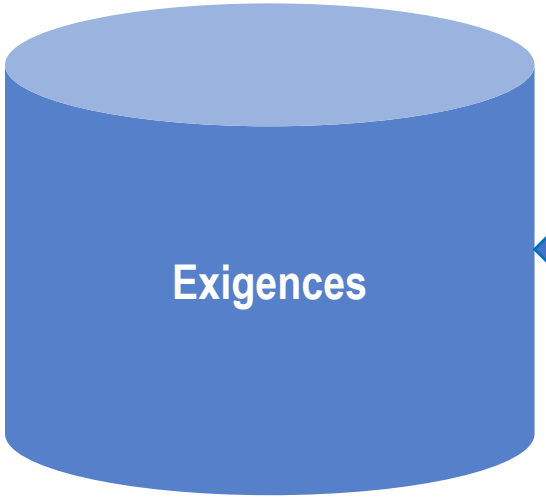
La traçabilité des exigences dans une approche basée modèle

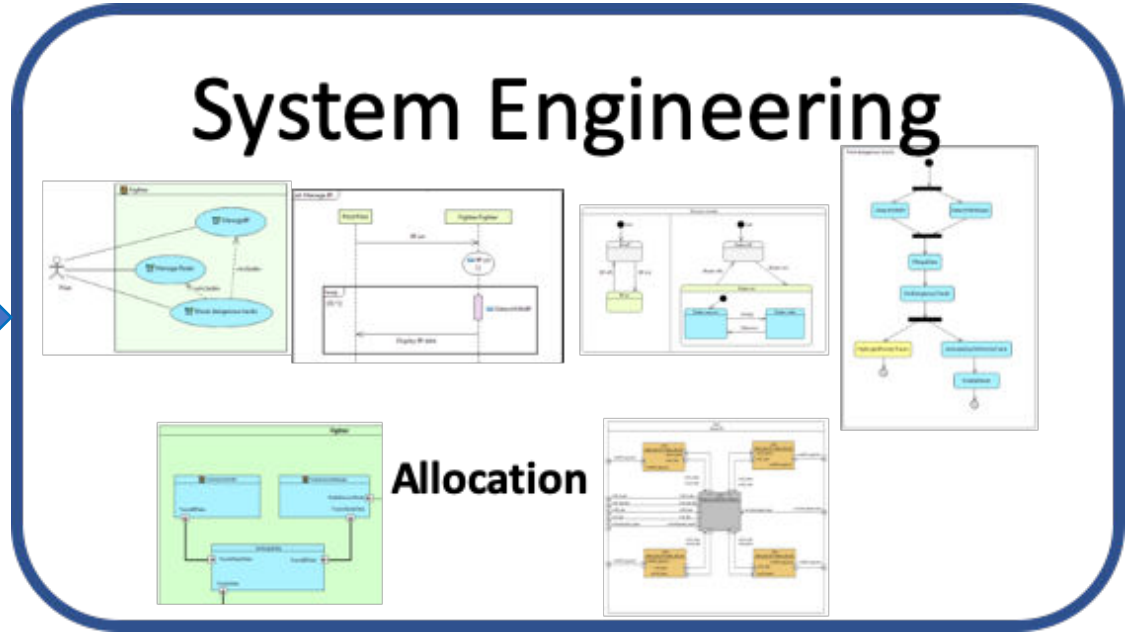


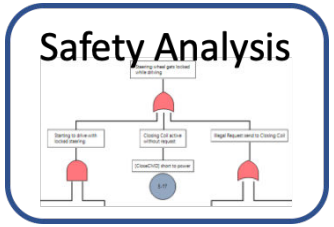
- Exigences des normes
  - Traçabilité bi-directionnelle exigences / conception / implémentation
  - Synchronisation exigences / conception
  - Traçabilité avec les artefacts de la vérification (tests, procédures, etc.)
- Analyse et vérification des exigences
  - Niveau de sûreté de fonctionnement
  - Couverture
- Mise à jour des exigences et exigences dérivées

Traçabilité Exigences / Modèles
Ingénierie Système





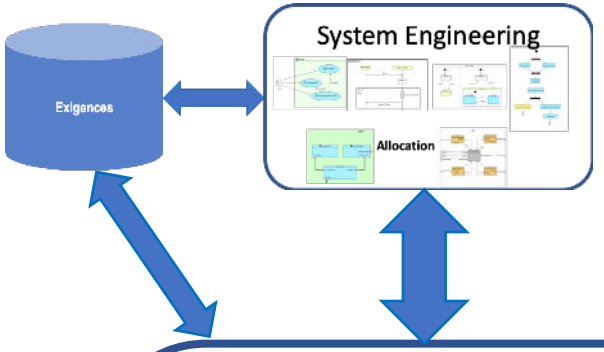




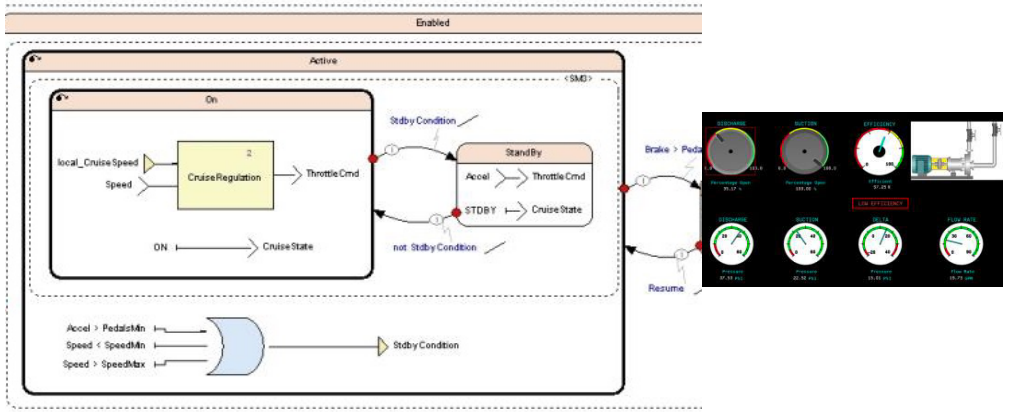
- Création, mise à jour
- Import de modèles
- Liens de traçabilité
- Matrices et analyse d'impact

- Conception et simulation de modèle d'architecture
- Lien de traçabilité
- Analyse de couverture, sûreté

Traçabilité Exigences / Modèles
Ingénierie Logicielle



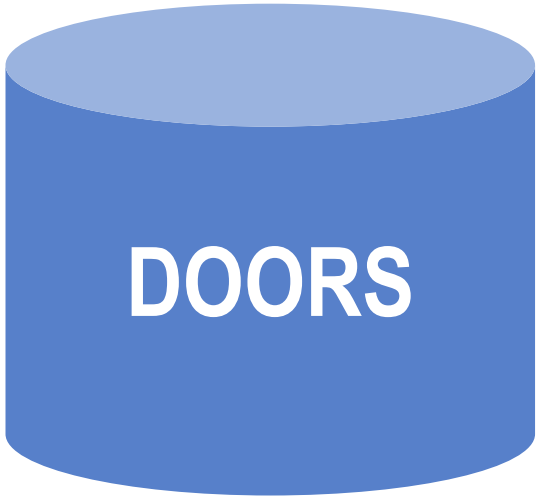
Ingénierie Logicielle

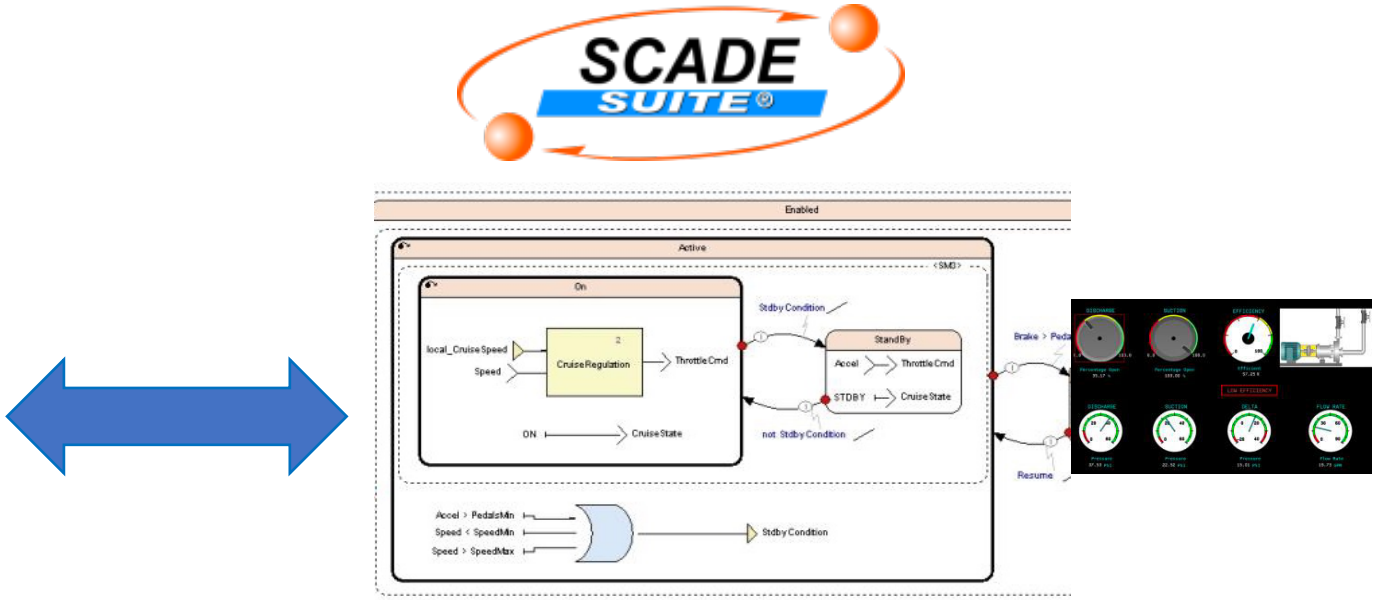


- Import de modèles et traçabilité système / logicelle automatique
- Liens de traçabilité
- Simulation, tests et verification formelle
- Couverture structurelle et fonctionnelle
- Génération automatique de code et traçabilité

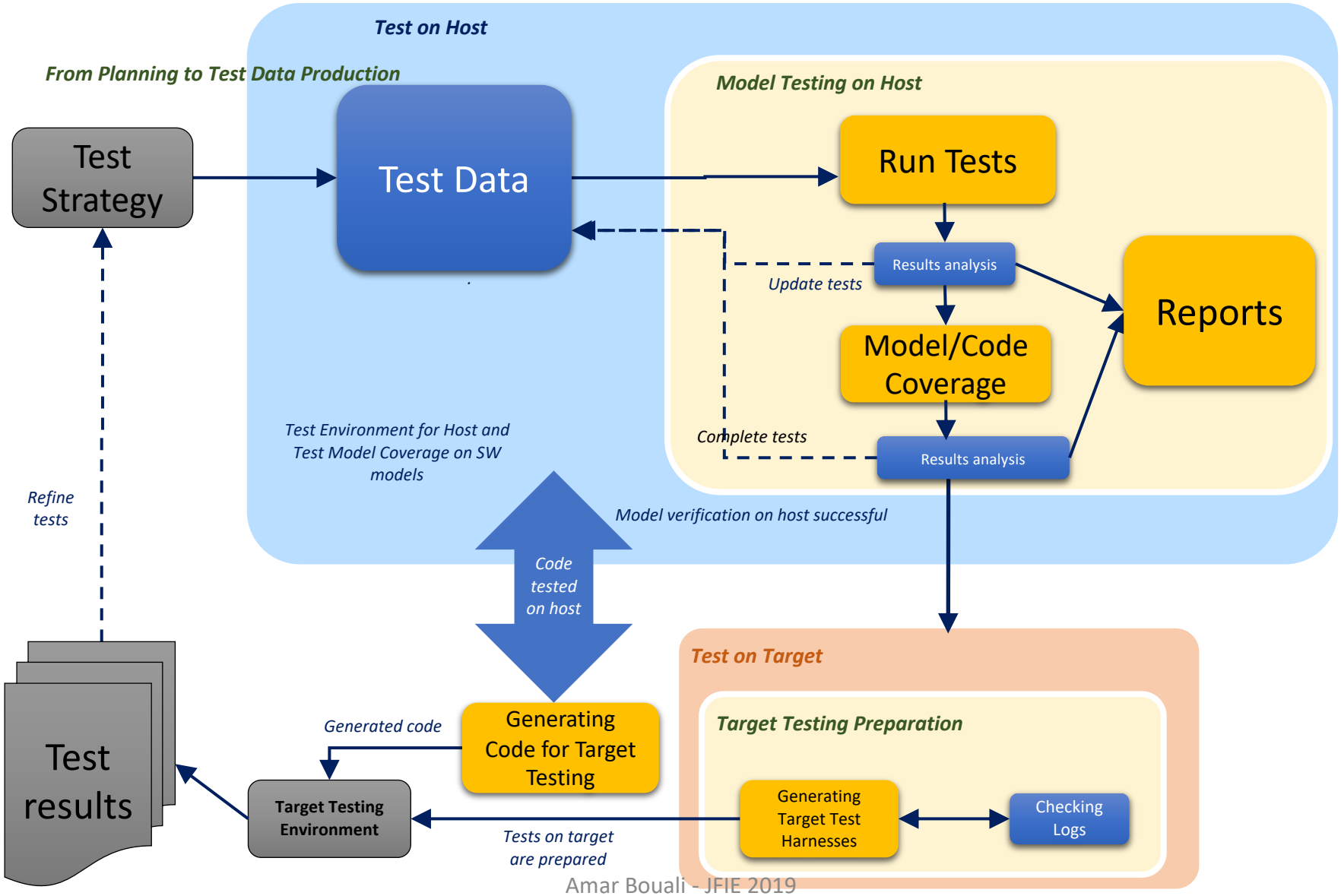
Exemple de solution industrielle

Traçabilité des modèles logiciels

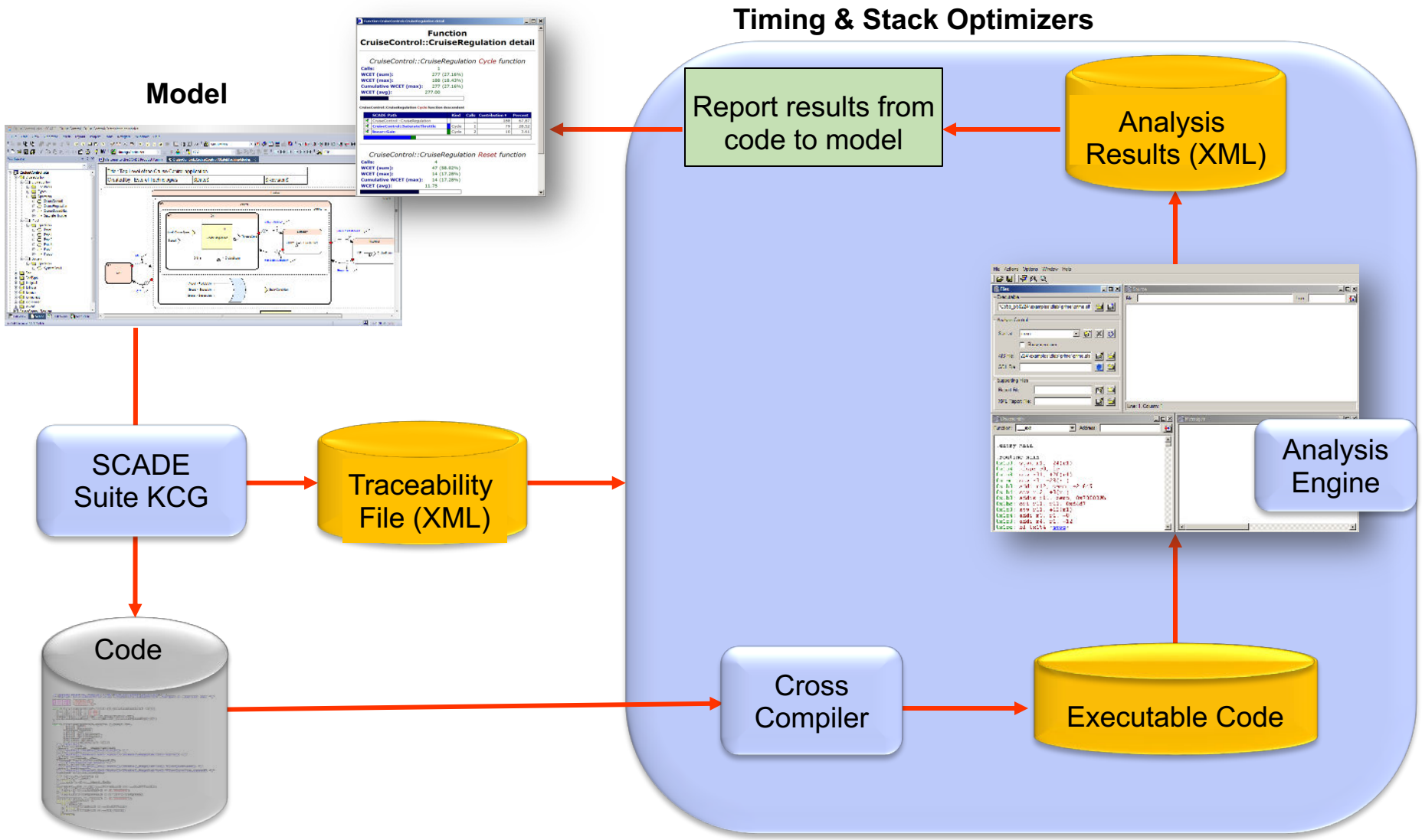




Traçabilité code objet ←→ modèle



Traçabilité code objet ←→ modèle



Conclusion

Bénéfices du basé-modèles

Consolidation des exigences systèmes

- Facilite la mise en oeuvre et la mise à jour des exigences
- Liens de traçabilité bidirectionnels Exigences / Modèles
- Augmente la détection d'erreurs et d'incomplétudes

Synchronisation ingénierie système / ingénierie logicielle

- Import / Export de modèles et traçabilité automatique accélérant la mise en œuvre
- Simulation, tests, vérification et vérification de la vérification basés modèles
- Cohérence constant entre modèles, et entre modèles et exigences

Automatisation

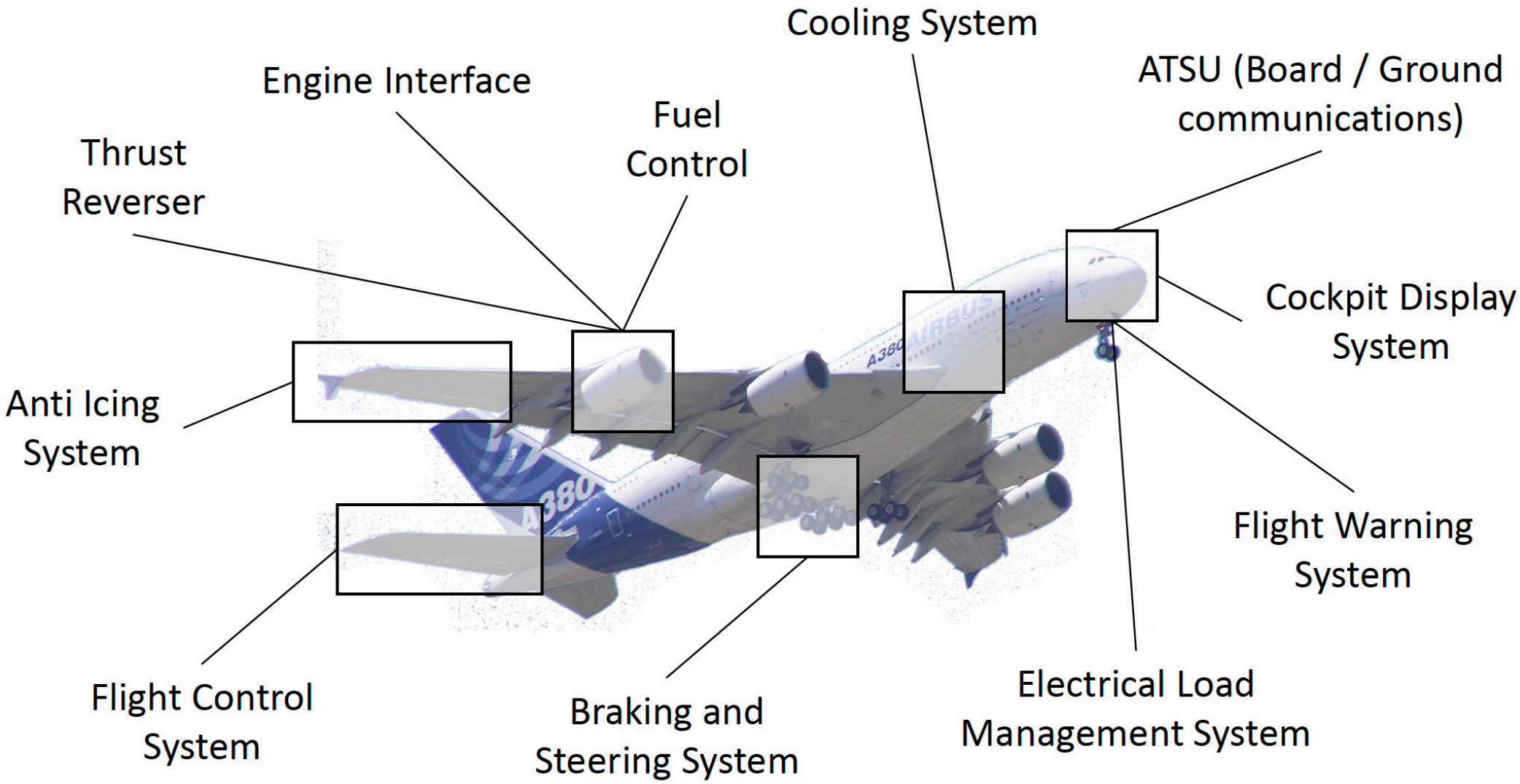
- Génération de code et de liens de tracabilité
- Génération de rapports à partir des modèles avec liens de traçabilité
- Génération de matrices de couverture et rapport de vérification

Backup

SCADE in the Airbus A380

AIRBUS

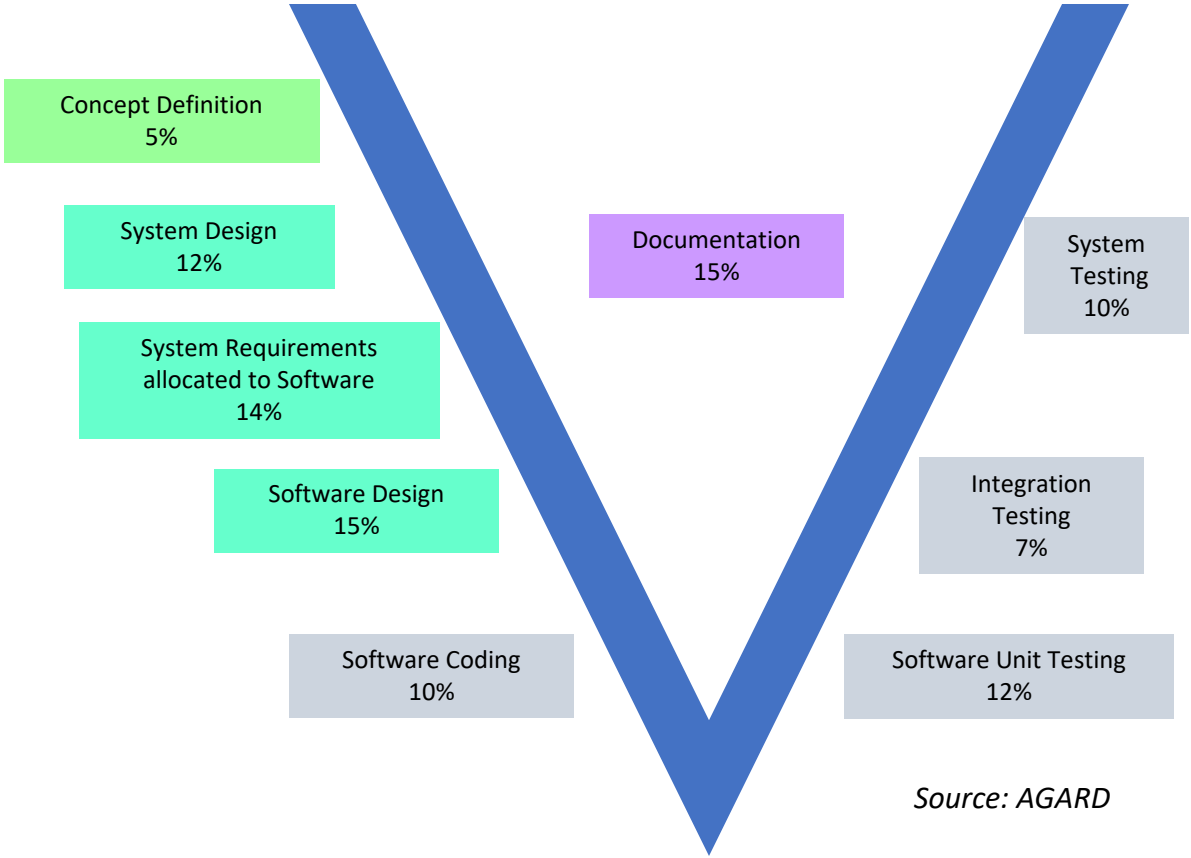
8 Million Lines of Code Generated !



Workload Breakdown on Aerospace Projects

Phase	*Effort (*)*
Concept Definition	**5%**
System Design (System Requirements, System Architecture, Control Laws Algorithms)	**12%**
System Requirements allocated to Software	**14%**
Software Design (Detailed Specifications)	**15%**
Coding	**10%**
Software Unit Testing	**12%**
Integration Testing	**7%**
System Testing	**10%**
Documentation	**15%**
TOTAL :	**100%**

(*) in Person x Months



Source: AGARD

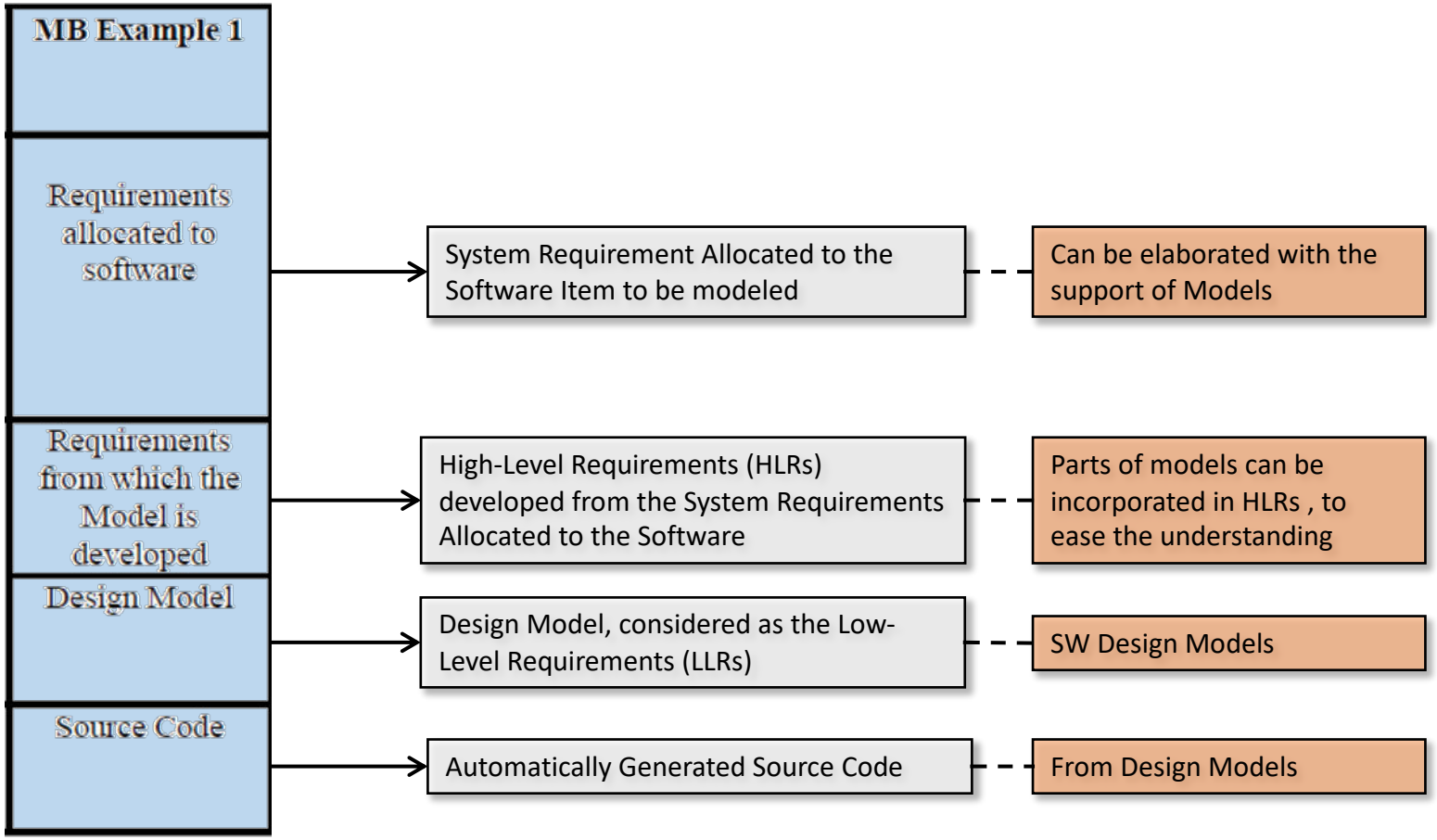
Cost Cutting Effect (Up to TQL)

Phase	Comments	Reference Cost Breakdown (Manual process)	Cost Breakdown (SCADE-based process)	SCADE Gain	Savings
Concept Definition		**5**	**5**	**0%**	Out of the scope of SCADE
System Design (Requirements, Functions and System Architecture)	Functional & Architectural Definition, System Safety Analysis	**12**	**8**	**35%**	Usage of SCADE System to model functions and architecture
System Requirements allocated to Software (HLRs)	Control Laws, Logic definition, HLRs (text, equations…)	**14**	**9**	**35%**	Reuse of functional and architectural definitions done in SCADE System
Software Design (LLRs)	Detailed SW architecture, Functional design, Requirements-based tests creation	**15**	**18**	**-20%**	Detailed SW architecture, Functional design (if not using SCADE for Control laws). Additionnal formalisation of Software detailed specifications, requirements traceability
Coding	Detailed Coding	**10**	**2**	**85%**	Percentage of code automatically generated with SCADE
Software Unit Testing (Low Level testing)	Functional Unit testing	**10**	**2**	**85%**	Qualification of the Code Generator suppresses low-level testing against the code generated with SCADE
Software / Software Integration & Testing	Testing of the above	**7**	**1**	**85%**	SW/SW Integration testing fully automated by SCADE for the SW application part
Hardware/Software Integration & Testing	incl. On target debugging	**10**	**5**	**50%**	Model already debugged - Very short late changes cycles, Compiler Verification Kit automates User Context verification and SCADE LifeCycle QTE automates application testing on target
Documentation & Reviews	Design documentation and Quality reviews	**7**	**1**	**85%**	Doc for project and for authorities is automatically generated by SCADE LifeCycle
Project Management		**10**	**5**	**50%**	Automation of connection with Config.Management Tools, Shortening of project duration, better requirements traceability thanks to SCADE LifeCycle
TOTAL :		**100**	**50**	**50%**	

Use of Model According to DO-178C/DO-331

Process that generates the life-cycle data	MB Example 1	MB Example 2	MB Example 3	MB Example 4 (See Note 1)	MB Example 5 (See Note 1)
System Requirement and System Design Processes	Requirements allocated to software	Requirements from which the Model is developed	Requirements from which the Model is developed	Requirements from which the Model is developed	Requirements from which the Model is developed
					Design Model
Software Requirement and Software Design Processes	Requirements from which the Model is developed	Specification Model (See Note 2)	Specification Model	Design Model	
	Design Model	Design Model	Textual description (See Note 3)		
Software Coding Process	Source Code	Source Code	Source Code	Source Code	Source Code

Use of Model According to DO-178C/DO-331

Process that generates the life-cycle data	MB Example 1	MB Example 2	MB Example 3	MB Example 4 (See Note 1)	MB Example 5 (See Note 1)
System Requirement and System Design Processes	Requirements allocated to software	Requirements from which the Model is developed	Requirements from which the Model is developed	Requirements from which the Model is developed	Requirements from which the Model is developed
					Design Model
Software Requirement and Software Design Processes	Requirements from which the Model is developed	Specification Model (See Note 2)	Specification Model	Design Model	
	Design Model	Design Model	Textual description (See Note 3)		
Software Coding Process	Source Code	Source Code	Source Code	Source Code	Source Code

Approche basée modèles