



Optimiser vos temps de développement et la qualité de vos produits grâce au « model based design »

karine.rouelle@ni.com

mathieu.bregeon@ni.com

s.serot@sherpa-eng.com

d.grolleau@sherpa-eng.com

Mardi 18 novembre 2014

Optimiser vos temps de développement et la qualité de vos produits grâce au « model based design »

- Introduction
- Du besoin au code applicatif
- Prototypage rapide
- Génération de code embarqué
- Validation HIL / Validation du produit fini

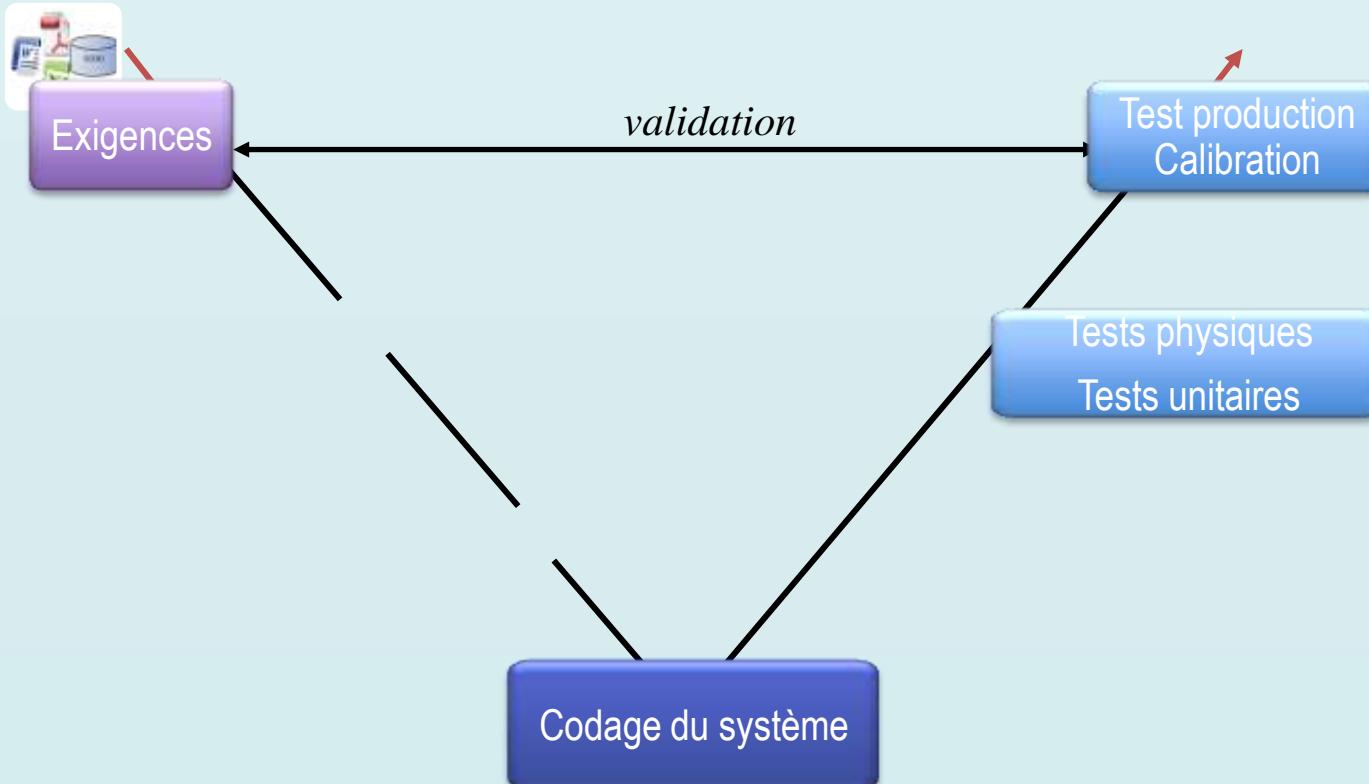


Optimiser vos temps de développement et la qualité de vos produits grâce au « model based design »

- **Introduction**
- Du besoin au code applicatif
- Prototypage rapide
- Génération de code embarqué
- Validation HIL / Validation du produit fini

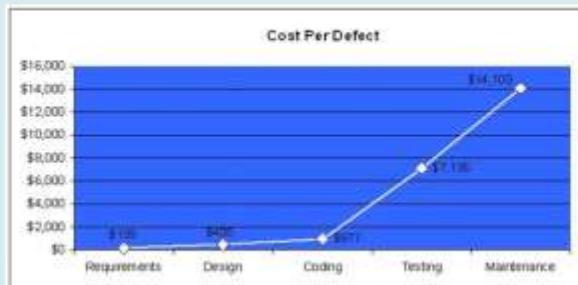


Introduction



Introduction

Machines grand public	Aéronautique	Automobile
		
100k 6.5 Mil 10 Mil		
10-20 défauts produits par 1,000 lignes de code*		
Défauts	1k – 2k	65k - 130k
100k – 200k		

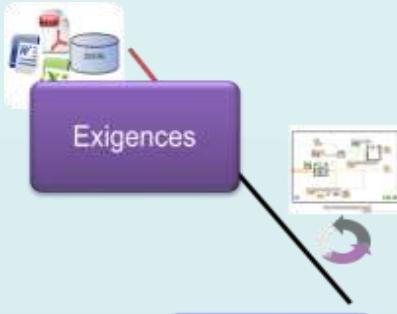


* The Economist May 16th, 2010. Tech View: Cars and software bugs. www.economist.com



Gérer le coût des erreurs logicielles

Introduction



Continuité
Validation
des tests

Modélisation
système

Lab
Tests véhicules

Module de test

Tests
physiques
Tests unitaires

Hardware-In
the-Loop (HIL)

Génération
automatique de
code, ECU
Flashing

Cycle de développement
des systèmes embarqués

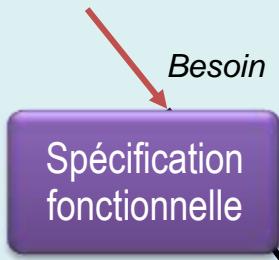


Optimiser vos temps de développement et la qualité de vos produits grâce au « model based design »

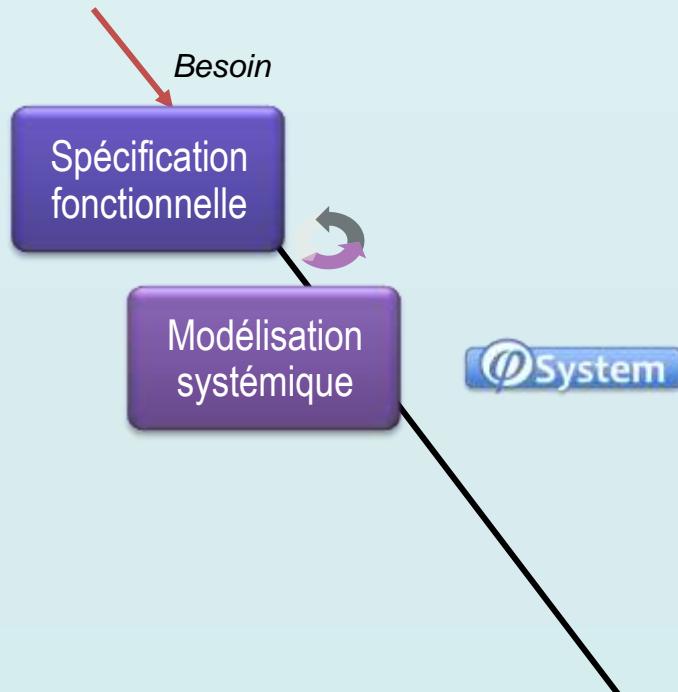
- Introduction
- **Du besoin au code applicatif**
- Prototypage rapide
- Génération de code embarqué
- Validation HIL / Validation du produit fini



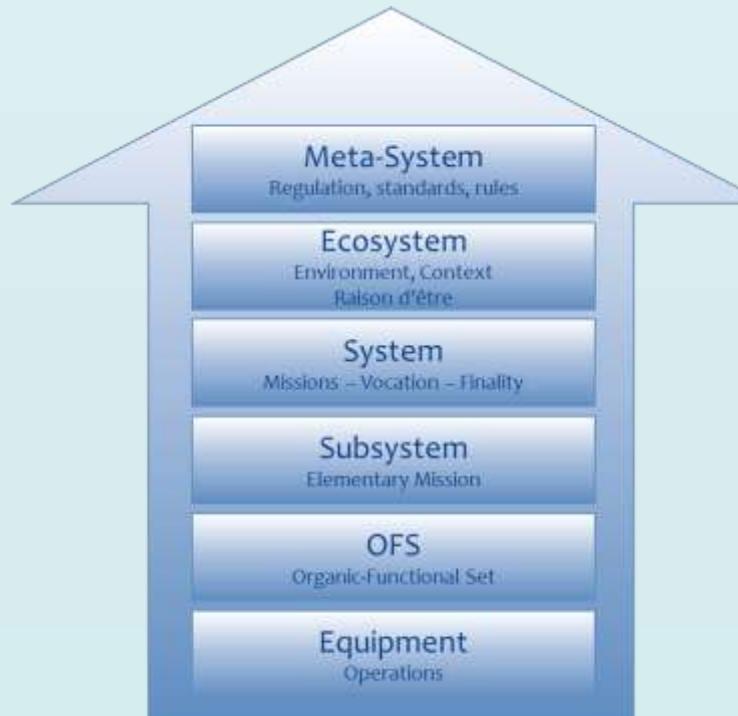
Du besoin au code applicatif



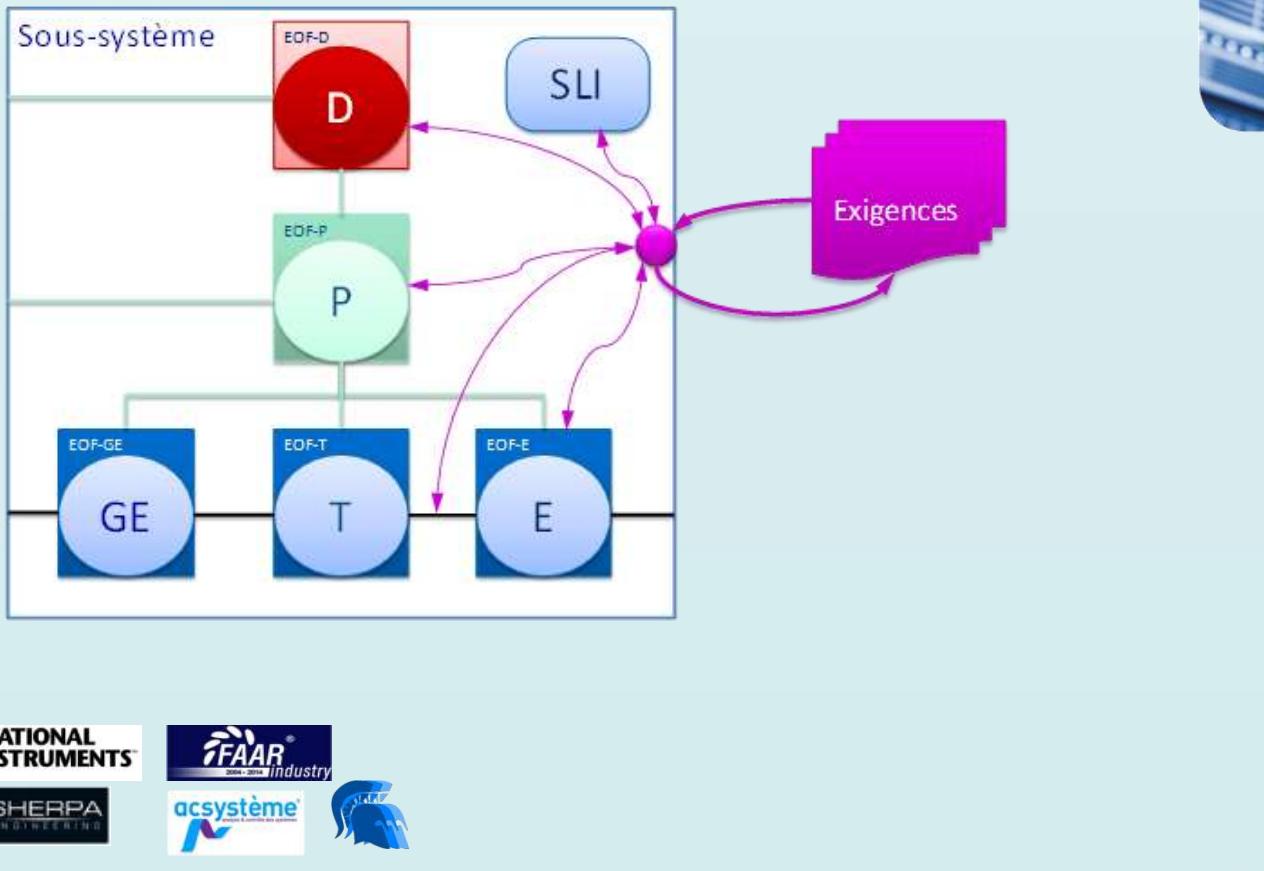
Du besoin au code applicatif



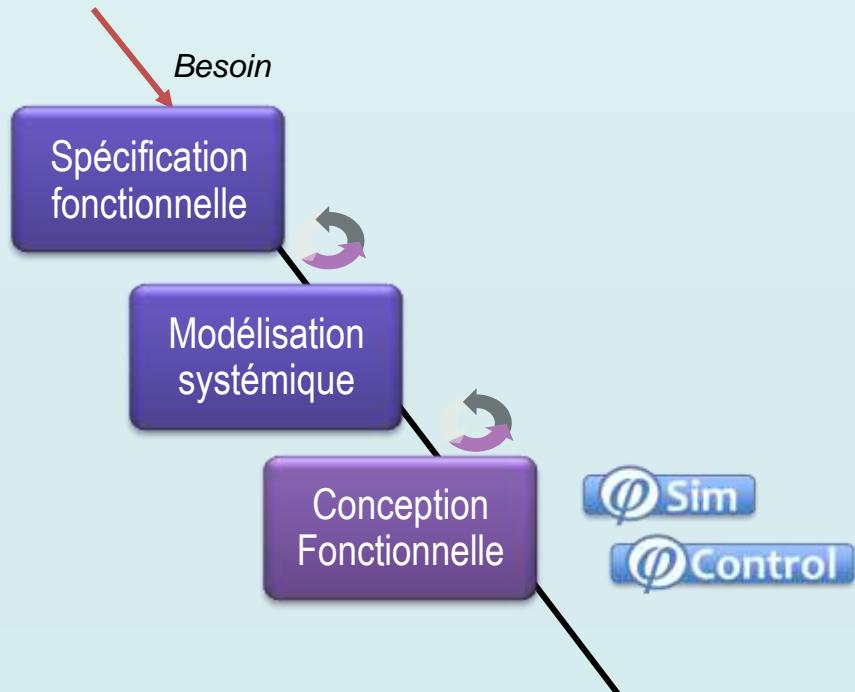
Du besoin au code applicatif



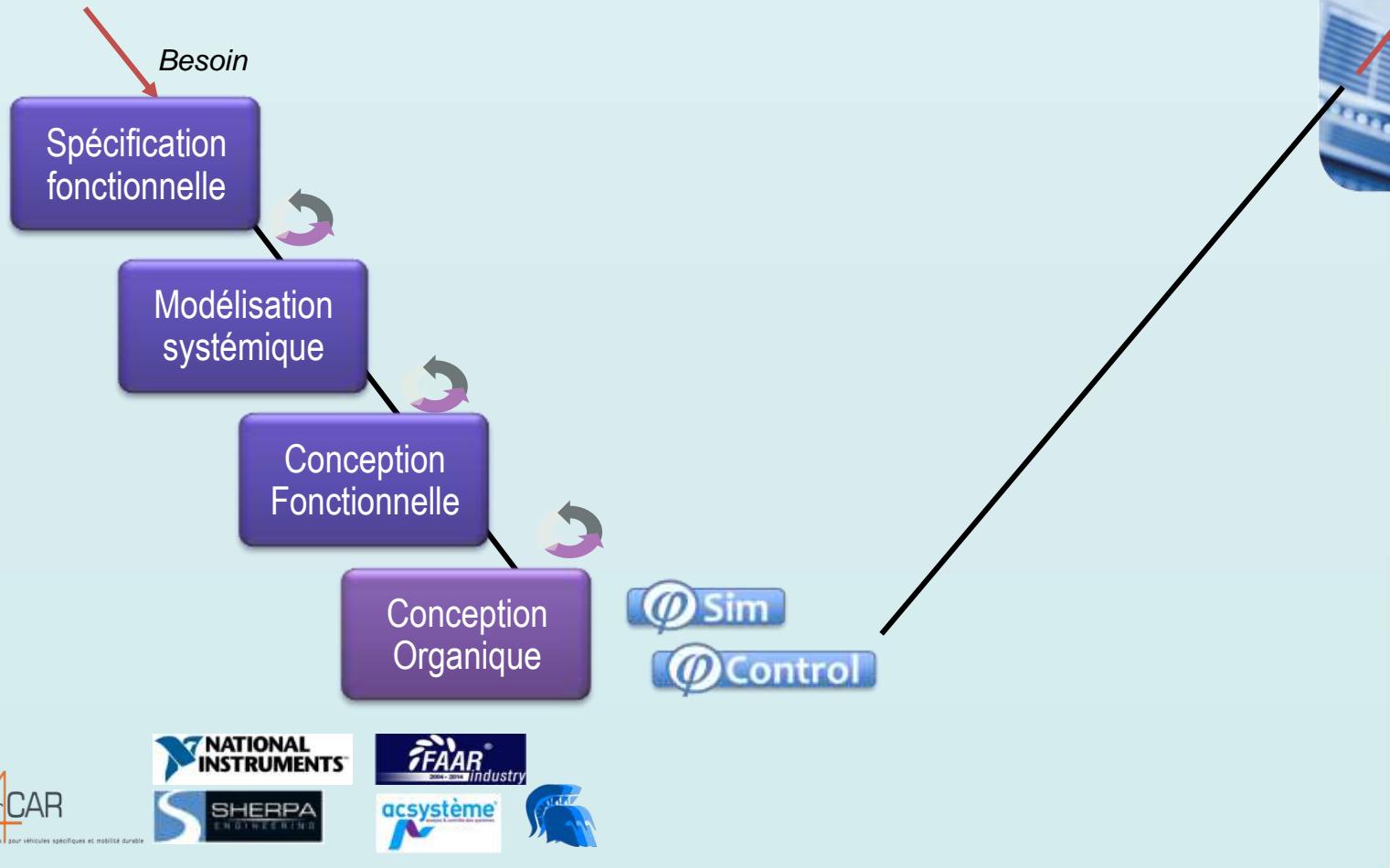
Du besoin au code applicatif



Du besoin au code applicatif



Du besoin au code applicatif



Du besoin au code applicatif

Définition

Analyse du besoin

Spécification fonctionnelle

Définition des exigences

Modèle systémique (ex : SysML)

- ▷ Diagramme de comportement
- ▷ Diagramme des exigences
- ▷ Diagramme de structure

Conception fonctionnelle

Conception organique

Vérification et Validation (V&V) à chaque étape

Traçabilité





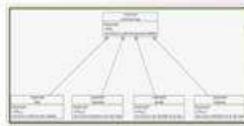
Les outils pour le « MBD »

	Définition Exigences	Gestion Exigences	Modélisation Systémique	Simulation Contrôle	Simulation multi-physics
SmarTeam	•	•	•		
Reqtify	••	•••			
DOORS	•••	•			
Rhapsody	••	•	•••	•	
Papyrus (PhiSystem)	••	•	•••		
Artisan	••	•	•••	•	
Entreprise Architect	•	•	•••		
arkItect	•	•	••		
Sirius	•	•	••		
Simulink (PhiSym / PhiControl)	•			•••	•••
Simscap (PhiSym)				•	•••
Scilab/Xcos				••	••
SimulationX				••	•••
Dymola				••	•••
AMESim				••	•••

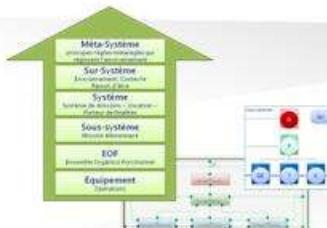


```
    gRgn.rgnCreateRectRgn(80,160,80+RectSizeL,160+RectSizeL);
    if(gRgn.rgnPtInRegion(point))
    {
        double x0c,y0c;
        for(int i=0;i<NUMCITY;i++)
        {
            x0c=80+RectSizeL*(iRectL)/NUMCITY;
            y0c=160+RectSizeL*(iRectL)/NUMCITY;
            CRgn rgnEllipse;
            rgnEllipse.CreateEllipticalRgn(x0c,y0c,RectSizeL,RectSizeL);
            if(gRgn.rgnIntersection(rgnEllipse))
                cout<<"Intersection found!"<<endl;
        }
    }
```

Collaborative Engineering Platform



Ingénierie des Exigences



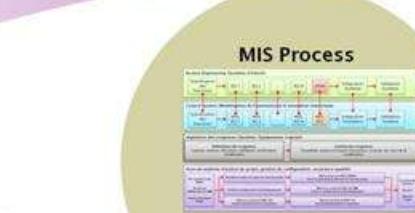
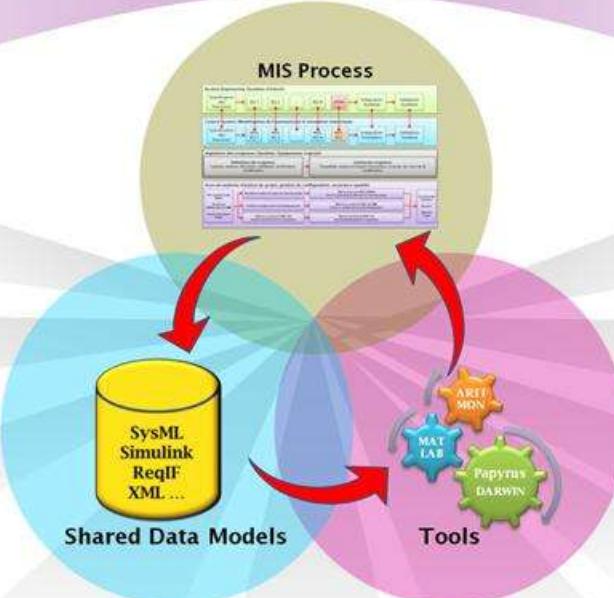
Modélisation Systémique Architecture



Modélisation & Simulations



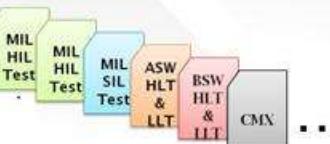
Collaborative Engineering Platform



V&V Système



Intégration, Validation HIL



Génération de tests

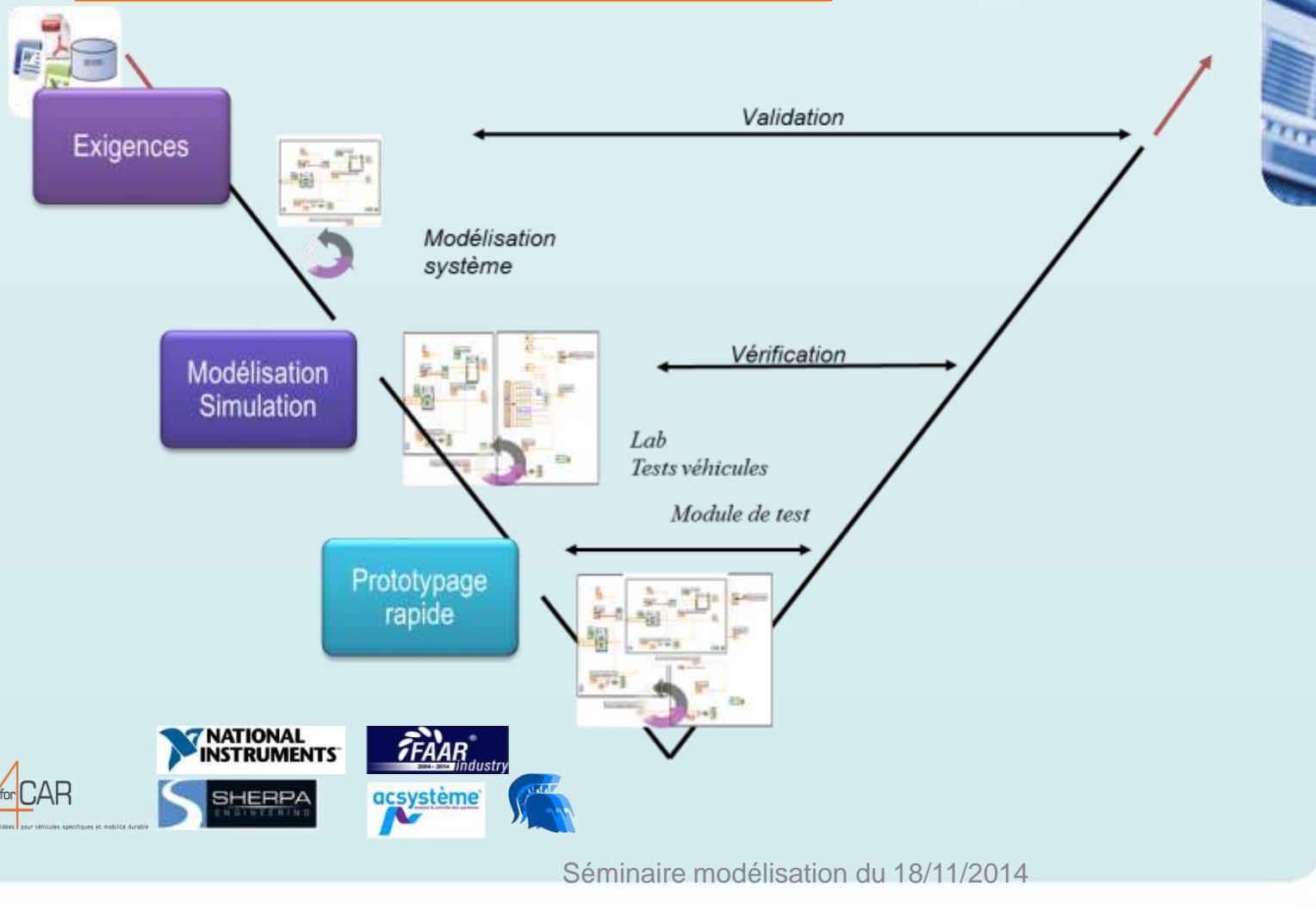


Optimiser vos temps de développement et la qualité de vos produits grâce au « model based design »

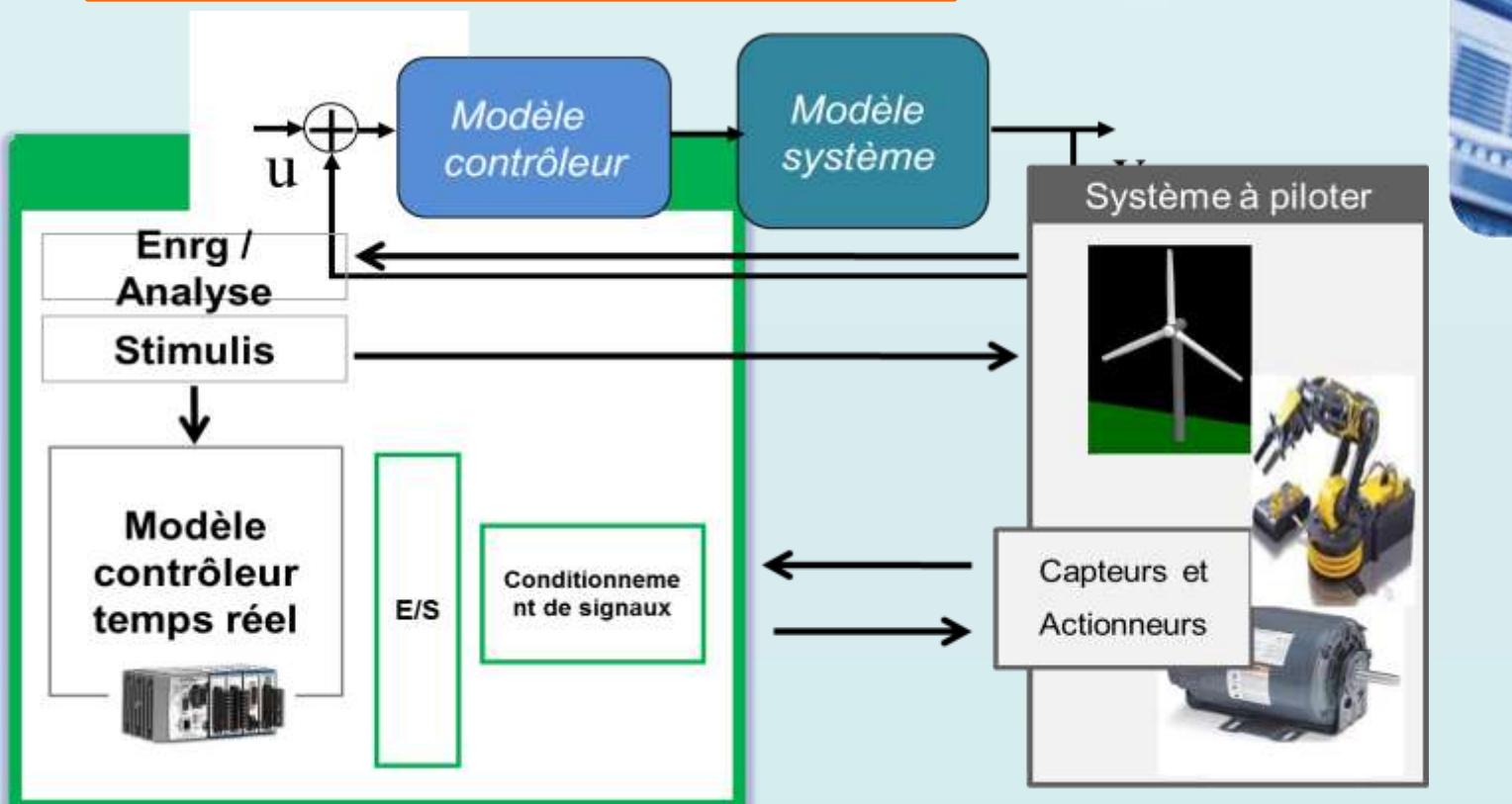
- Introduction
- Du besoin au code applicatif
- **Prototypage rapide**
- Génération de code embarqué
- Validation HIL / Validation du produit fini



Prototypage rapide



Prototypage rapide



```
Sign
CreateRectRgn(80,160,80+RectSizeX*NUMCITY,y0+RectSizeY*NUMCITY)
gdi.FillRegion(point)
double x0c,y0c;
for(int i=0;i<NUMCITY;i++)
{
    x0c=80+RectSizeX*i*10;
    y0c=160+RectSizeY*i*10;
    gdi.Ellipse(x0c,y0c,x0c+RectSizeX,y0c+RectSizeY);
    CreateEllipticRgn(x0c,y0c,x0c+RectSizeX,y0c+RectSizeY);
}
```

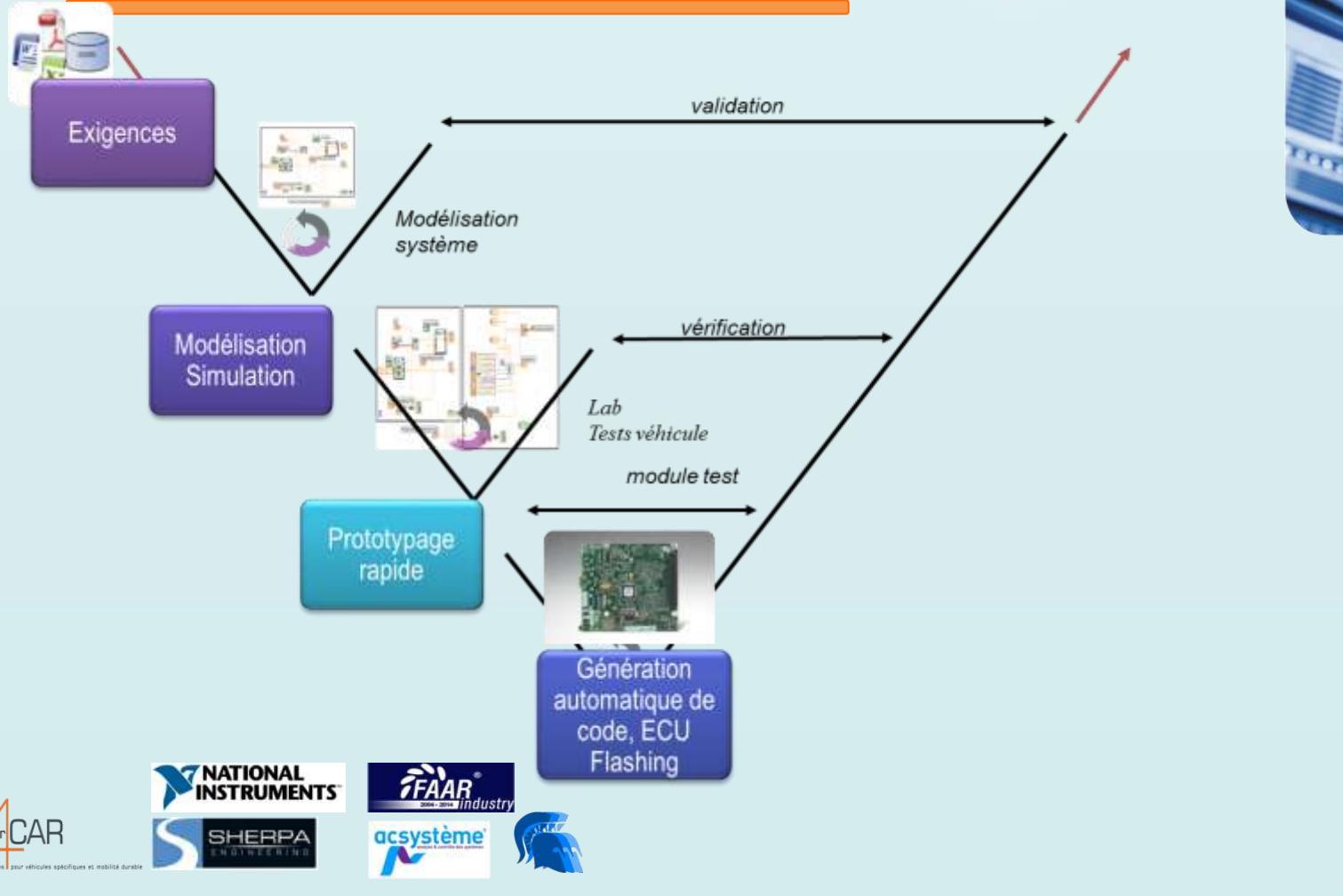


Optimiser vos temps de développement et la qualité de vos produits grâce au « model based design »

- Introduction
- Du besoin au code applicatif
- Prototypage rapide
- **Génération de code embarqué**
- Validation HIL / Validation du produit fini



Génération de code embarqué



Génération de code embarqué

Du modèle vers le SW applicatif

Adapter le modèle à la cible

- Continu → discret
- Virgule flottante → virgule fixe
- Adaptation du modèle (contraintes dues à la cible)
- Génération de code
- Code de validation
- Optimisation du code + couches bas niveau
- Code embarqué

Modèle contrôle commande

Modèle physique si simulation temps réel

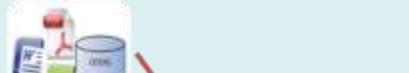


Optimiser vos temps de développement et la qualité de vos produits grâce au « model based design »

- Introduction
- Du besoin au code applicatif
- Prototypage rapide
- Génération de code embarqué
- **Validation HIL / Validation du produit fini**



Validation HIL



Exigences



Modélisation système

Modélisation Simulation



validation

vérification

Lab
Tests véhicules

Prototypage rapide



Hardware-In the -Loop (HIL)

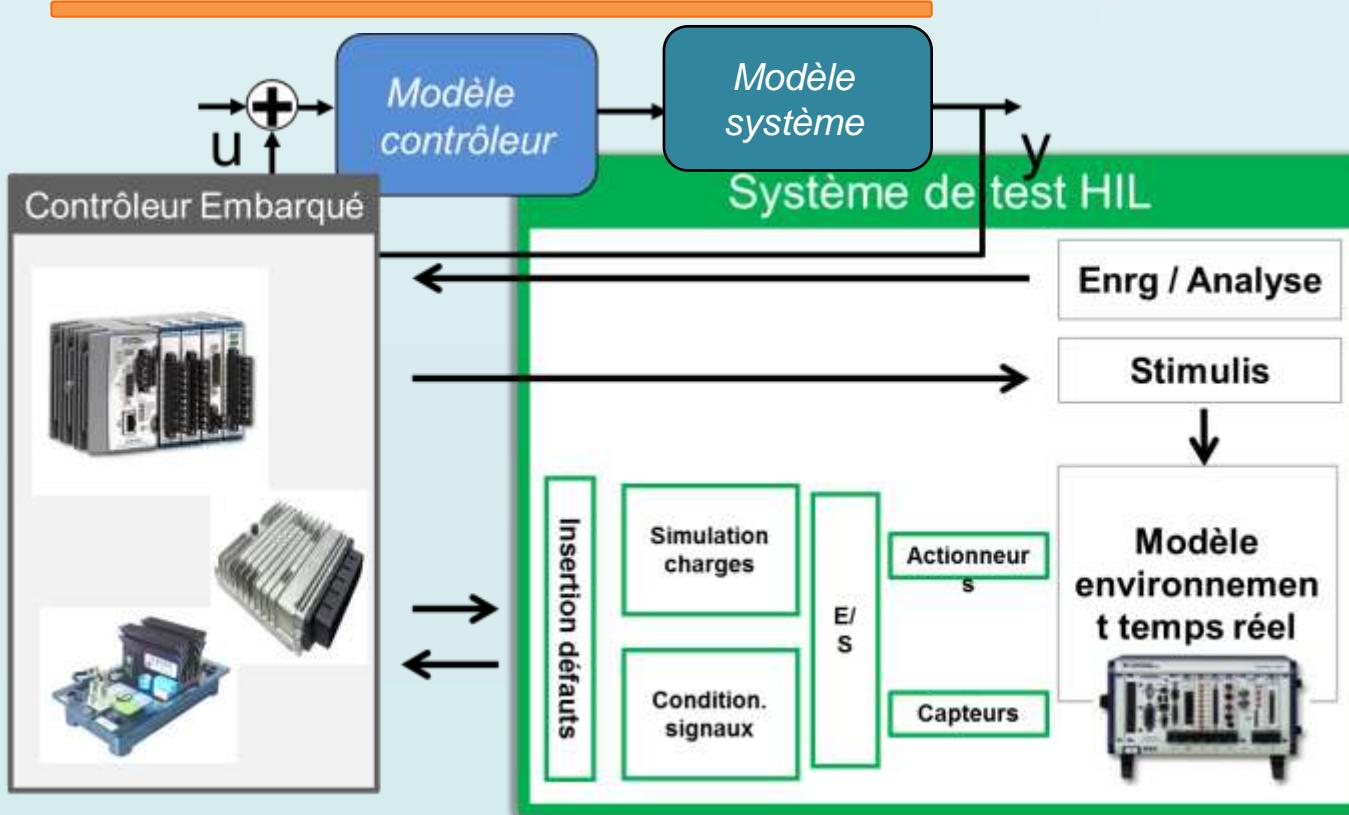
module de test

Génération automatique de code, ECU Flashing

```
rgn=CreateRectRgn(80,160,80+RectSizeX*i,160+RectSizeY*i)
for(int i=0;i<NUMCITY;i++)
{
    x0c=80+RectSizeX*i;
    y0c=160+RectSizeY*i;
    rgnEllipse=>CreateEllipticRgn(x0c,y0c,RectSizeX,RectSizeY);
    CreateRegion(rgn,rgnEllipse);
}
```



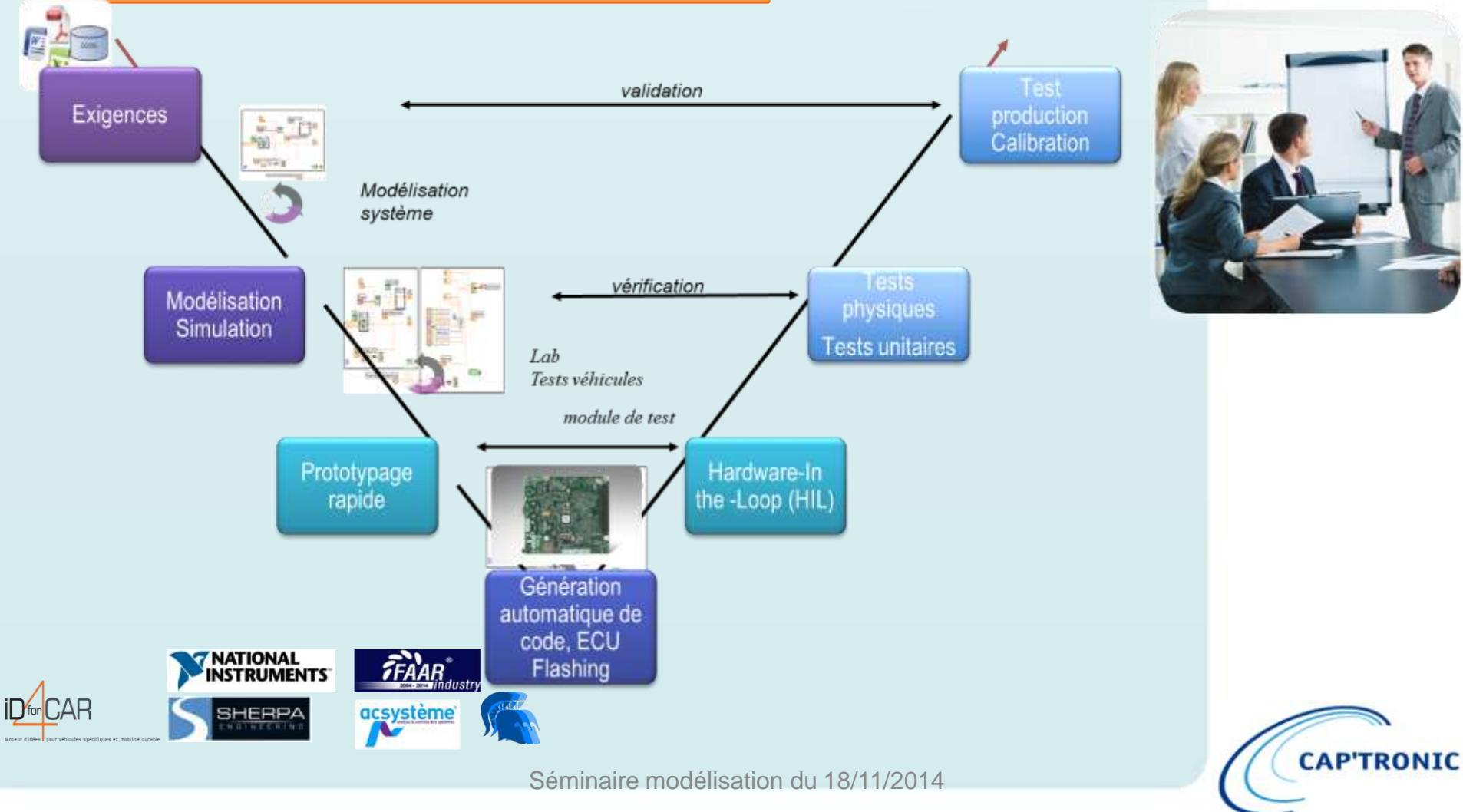
Validation HIL



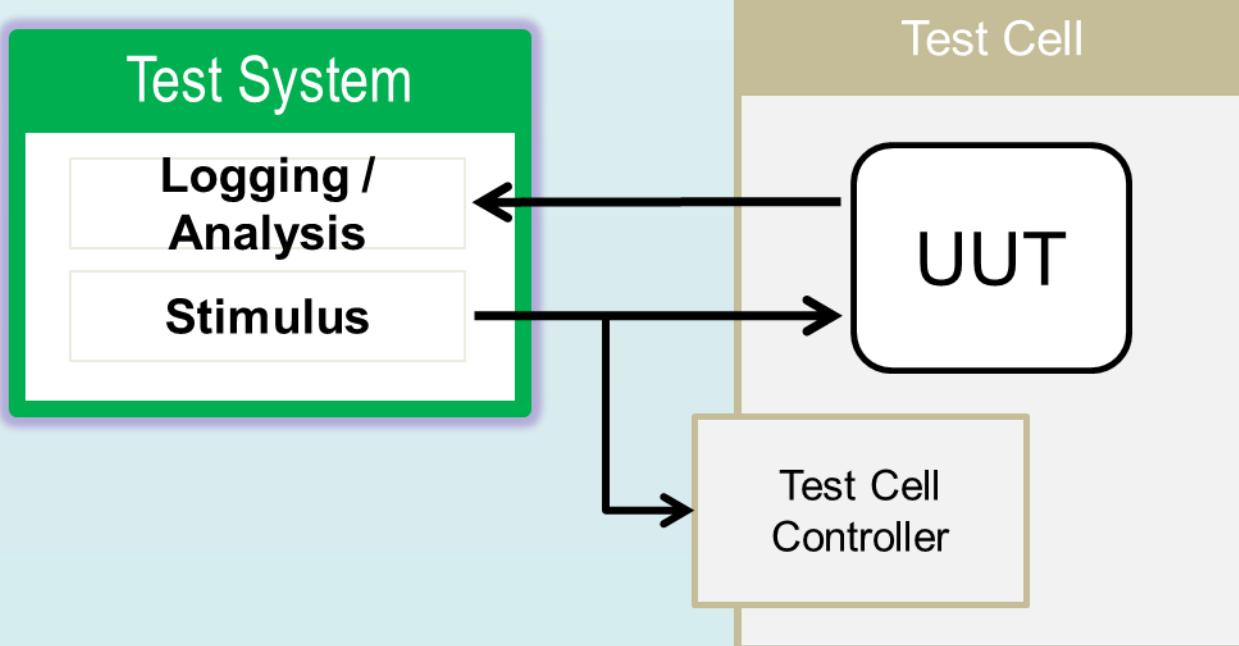
```
Sign
CreateRectRgn(80,160,80+RectSizeX*NUMCITY,y0+RectSizeY*NUMCITY)
gdi.FillRegion(point)
double x0c,y0c;
for(int i=0;i<NUMCITY;i++)
{
    x0c=80+RectSizeX*i*16;
    y0c=160+RectSizeY*i*16;
    gdi.Ellipse(x0c,y0c,x0c+RectSizeX,y0c+RectSizeY);
    CreateEllipticRgn(x0c,y0c,RectSizeX,RectSizeY);
}
```



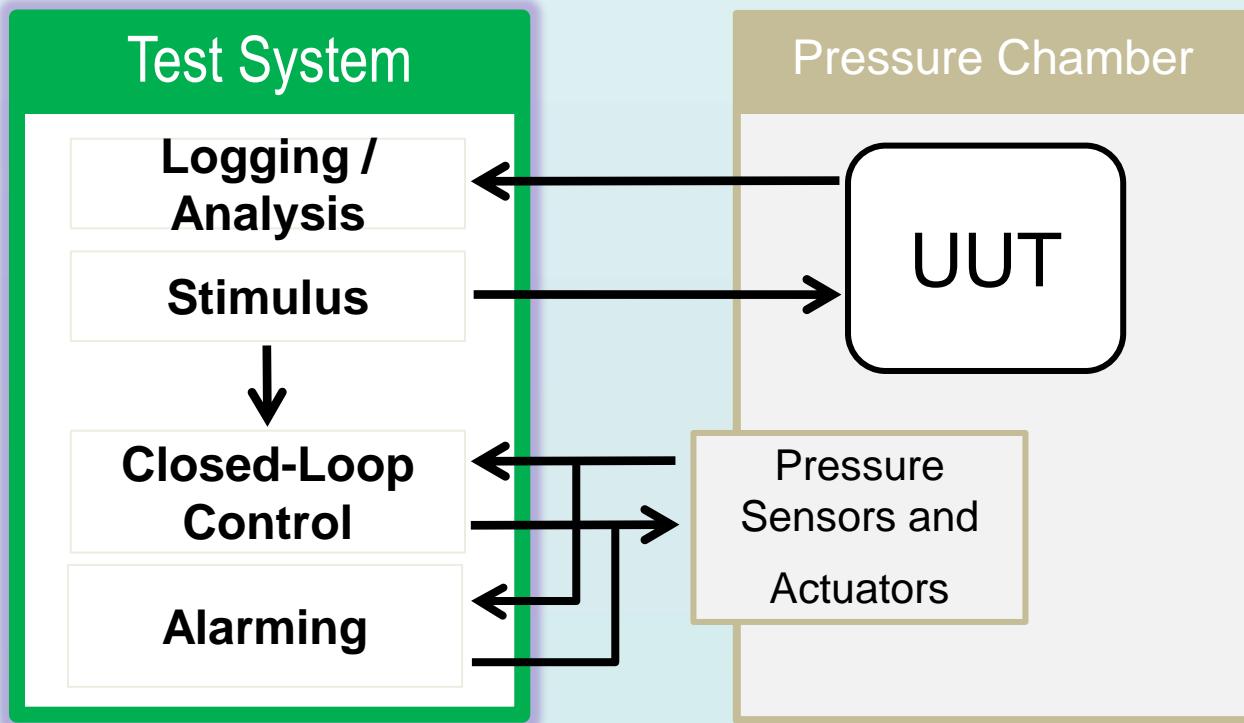
Validation du produit fini



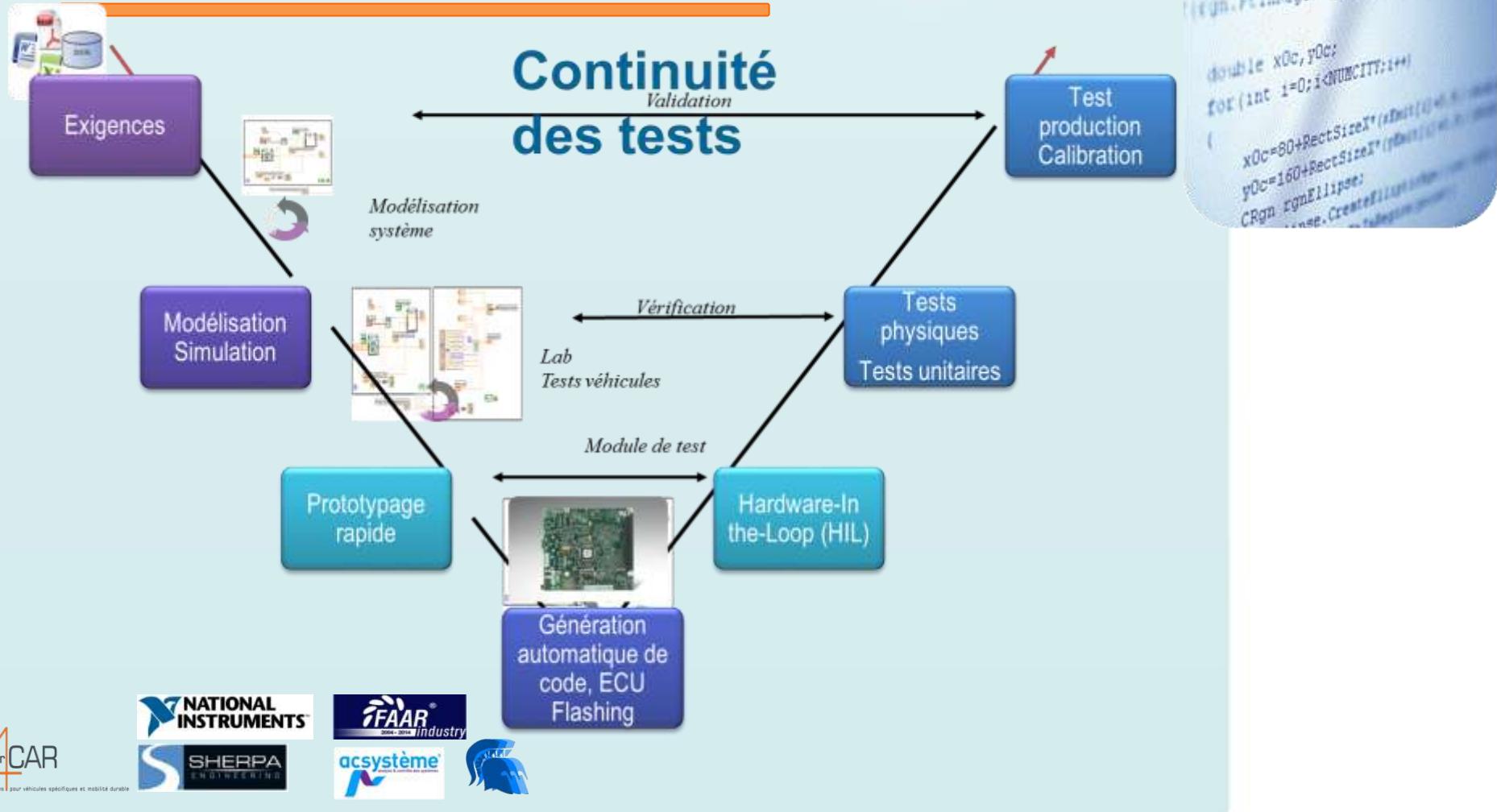
Validation du produit fini



Validation du produit fini



Cycle de développement orienté « MBD »



Conclusion





www.france.ni.com



www.sherpa-eng.com

> www.captronic.fr



Moteur d'idées pour véhicules spécifiques et mobilité durable



JESSICA FRANCE - 17, rue des Martyrs - 38054 GRENOBLE

Cedex 09

CAP'TRONIC, un programme financé par :

JESSICA FRANCE fondée par :

