



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

```
//gnc  
CreateRectPgn(80,160,80+  
gn.PtInRegion(point))  
  
void* xOc,yOc;  
for (int i=0;i<NUMCITY;i++)  
{  
    xOc=80+RectSizeX*(sEmt  
    yOc=160+RectSizeY*(yEmt  
    gnc.RgnEllipse:  
    - CreateEllipse
```



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



```
//gnc  
CreateRectPgn(80,160,80+  
gn, PtInRegion(point))  
  
void* xOc,yOc;  
for (int i=0;i<NUMCITY;i++)  
{  
    xOc=80+RectSizeX*(i%5)  
    yOc=160+RectSizeY*(i/5)  
    gnc.RgnEllipse:  
    - CreateEllipse
```

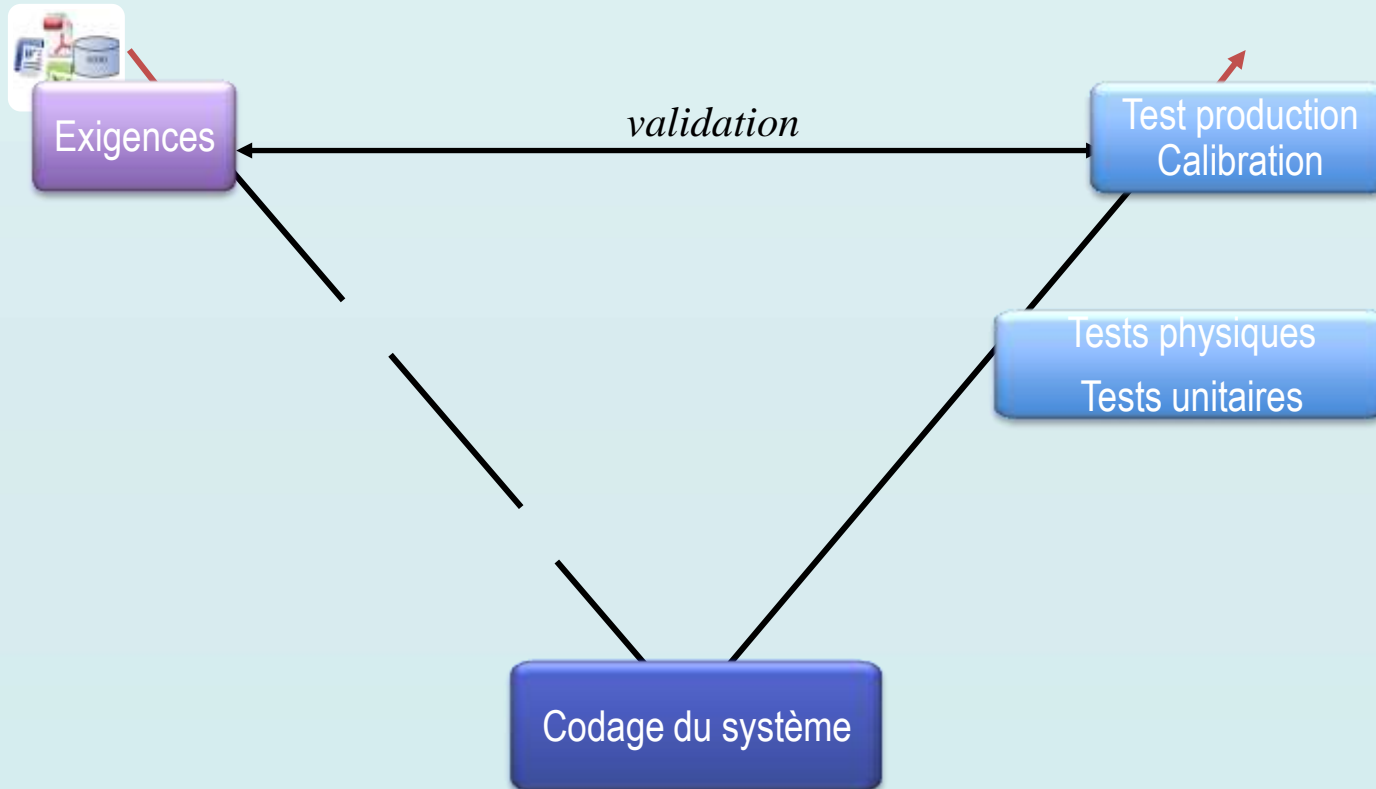


Optimiser vos temps de développement et la qualité de vos produits grâce au « model based design »

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



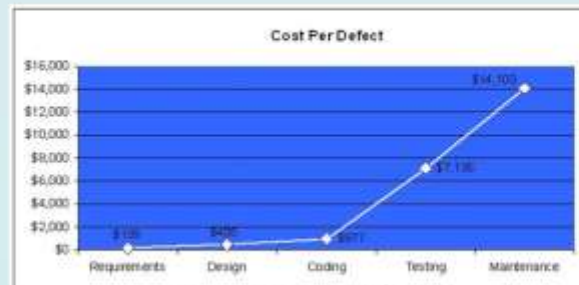
Introduction



Introduction



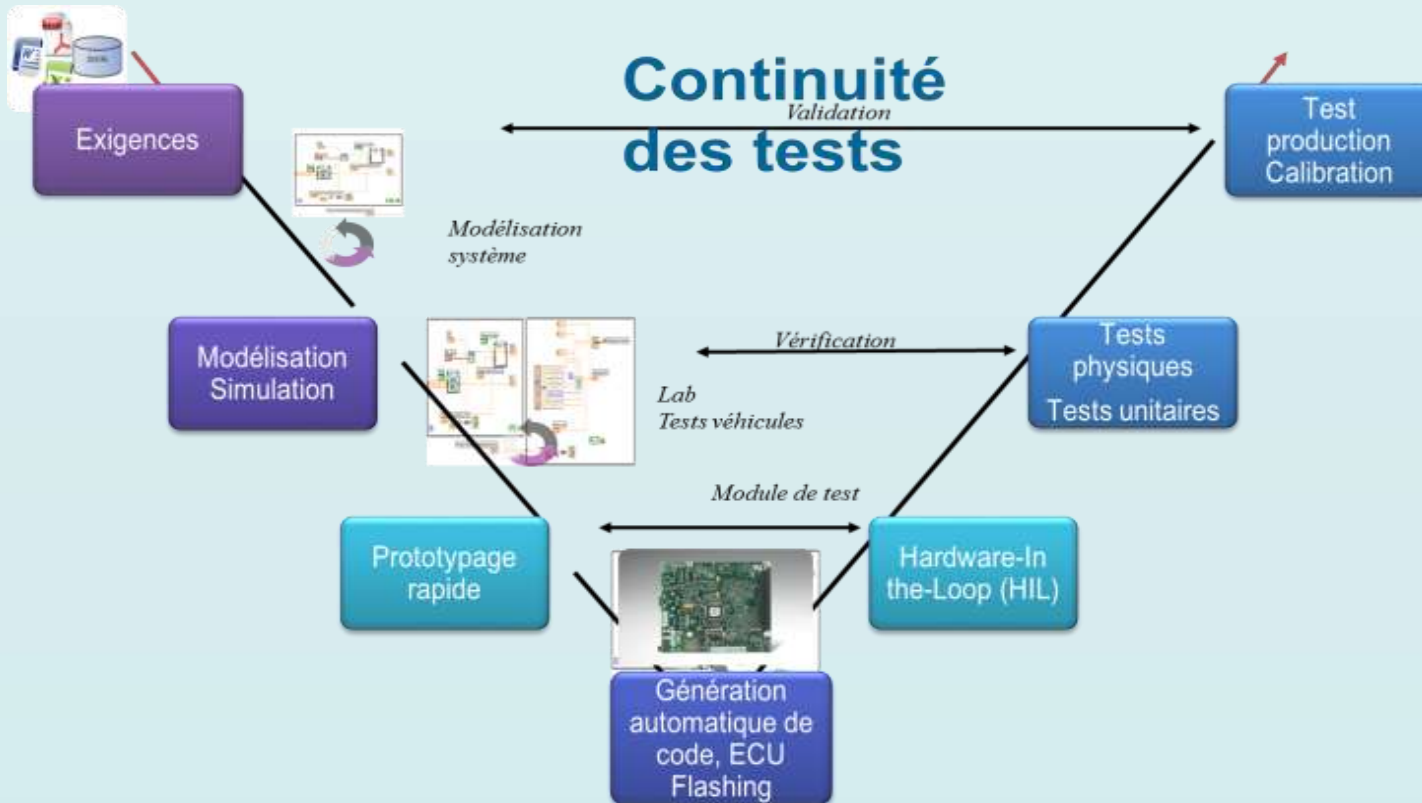
	Machines grand public	Aéronautique	Automobile
			
Nbre lignes de code	100k	6.5 Mil	10 Mil
	10-20 défauts produits par 1,000 lignes de code*		
Défauts	1k – 2k	65k - 130k	100k – 200k



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

Gérer le coût des erreurs logicielles

Introduction



Cycle de développement
des systèmes embarqués

```
//gnc  
CreateRectPgn(80,160,80+  
gn, PtInRegion(point))  
  
void* xOc,yOc;  
for (int i=0;i<NUMCITY;i++)  
{  
    xOc=80+RectSizeX*(sEmi  
    yOc=160+RectSizeX*(yEmi  
    gnc.RgnEllipse:  
    - CreateEllipse
```

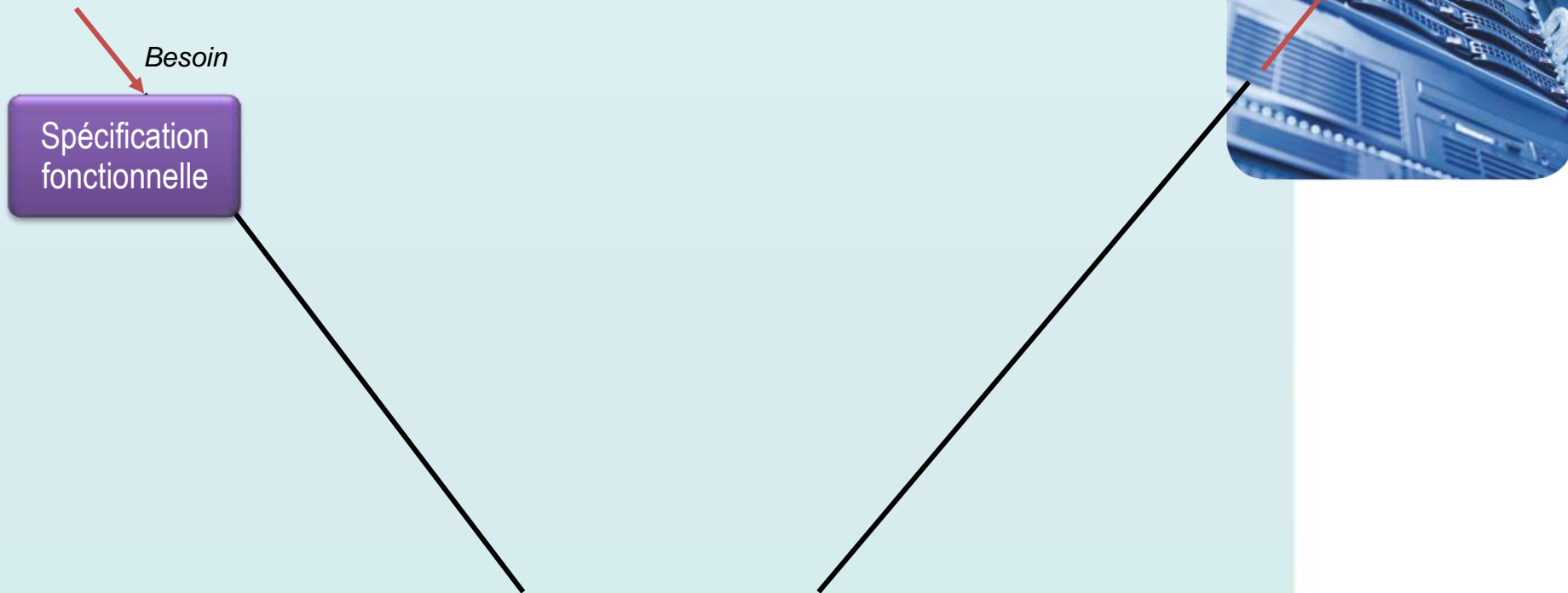


Optimiser vos temps de développement et la qualité de vos produits grâce au « model based design »

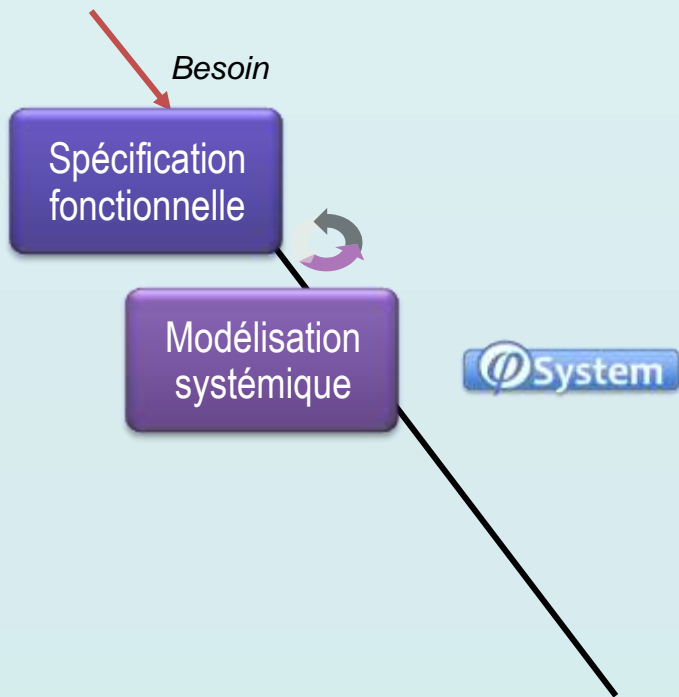
- Introduction
- **Du besoin au code applicatif**
- Prototypage rapide
- Génération de code embarqué
- Validation HIL / Validation du produit fini



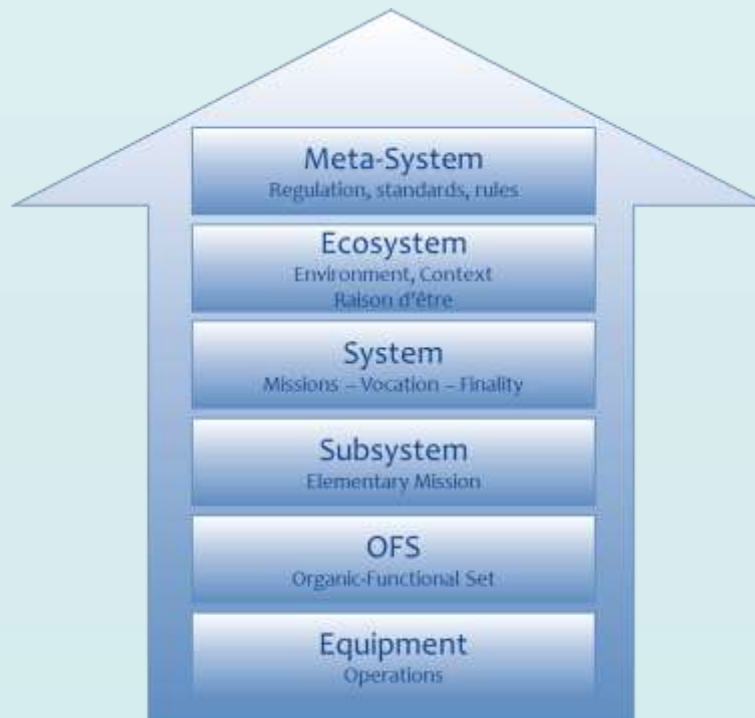
Du besoin au code applicatif



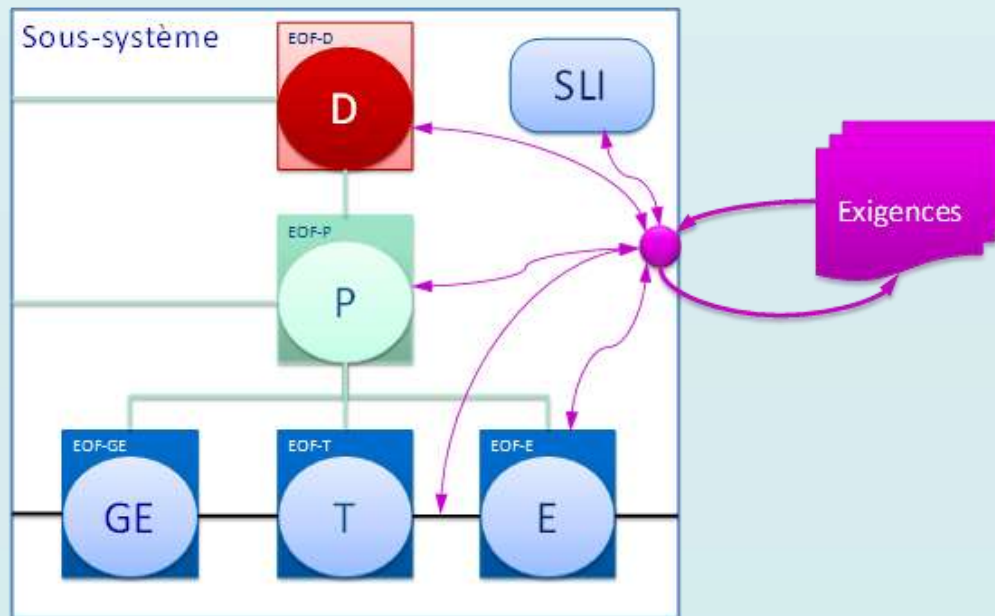
Du besoin au code applicatif



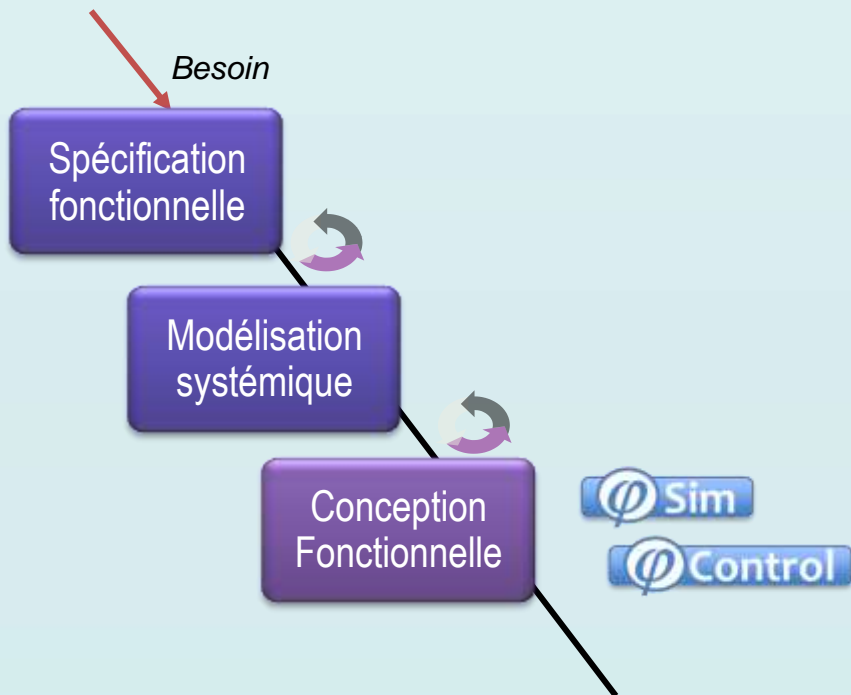
Du besoin au code applicatif



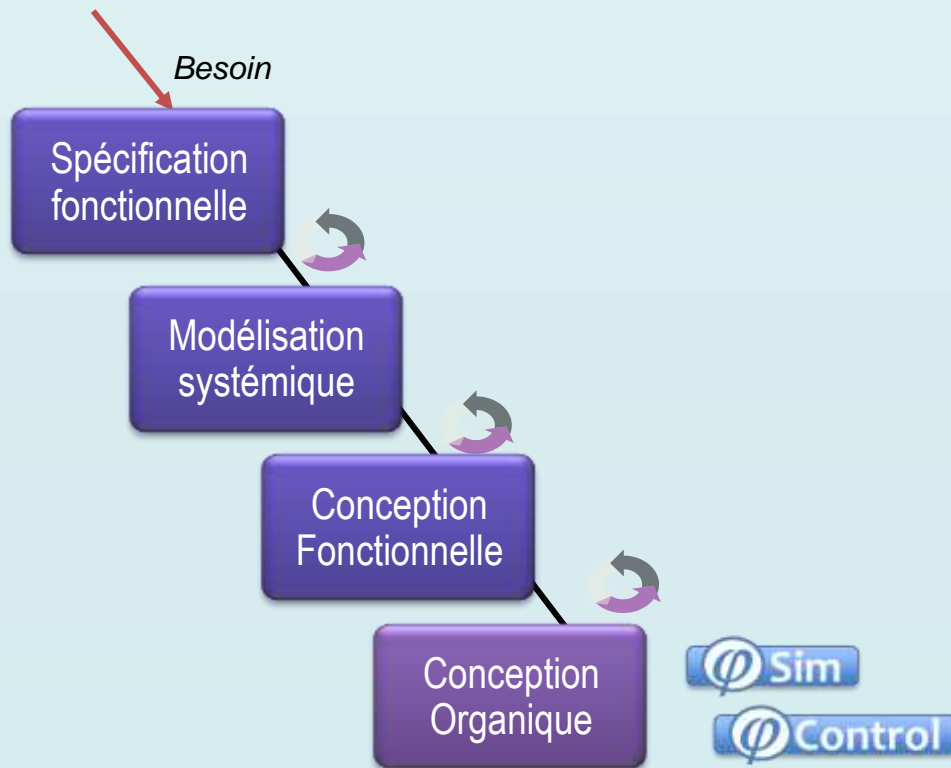
Du besoin au code applicatif



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

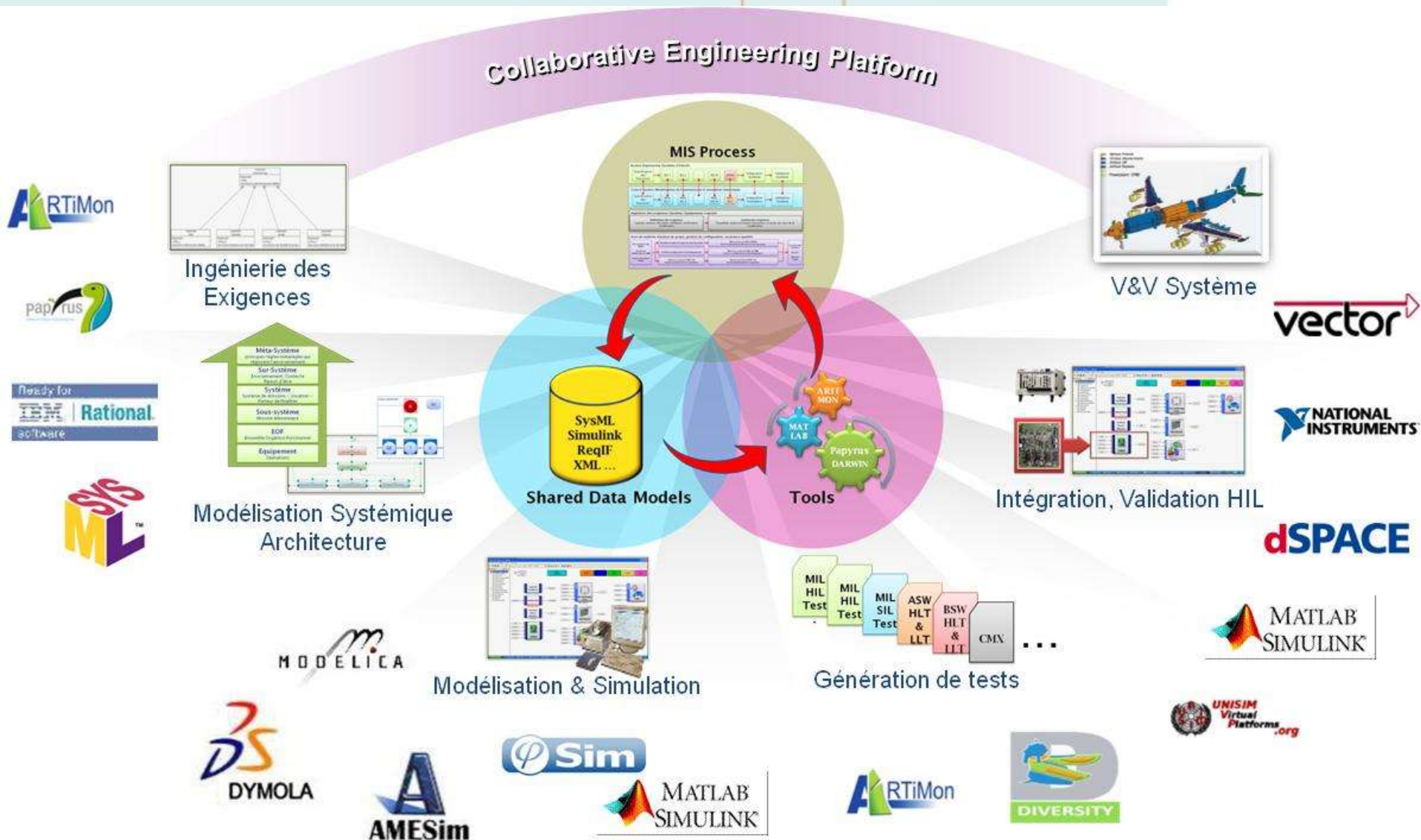


Les outils pour le « MBD »

	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				••	•••



Collaborative Engineering Platform



```
//gnc  
CreateRectPgn(80,160,80+  
gn.PtInRegion(point))  
  
voidie xOc,yOc;  
for (int i=0;i<NUNCITY;i++)  
{  
    xOc=80+RectSizeX*(sExi  
    yOc=160+RectSizeY*(yExi  
    gnc.RgnEllipse:  
    - CreateEllipse
```

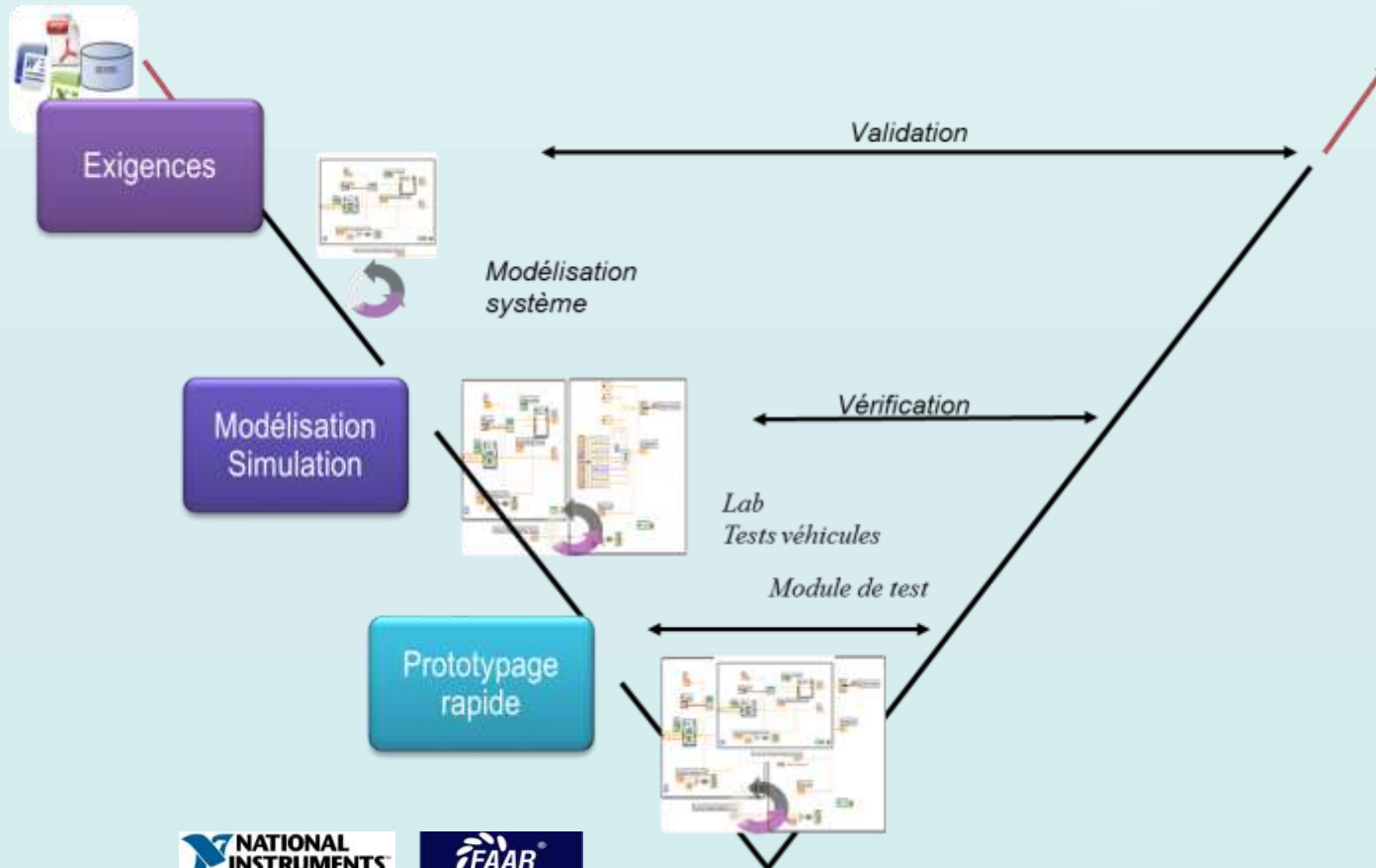


Optimiser vos temps de développement et la qualité de vos produits grâce au « model based design »

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



Prototypage rapide




```
gfnz  
CreateRectPgn(80,160,80+  
gn.FtInRegion(point))  
voidie xOc,yOc;  
for (int i=0;i<NUMCITY;i++)  
xOc=80+RectSizeX*(sExi  
yOc=160+RectSizeX*(yExi  
-gn.FgnEllipse;  
- CreateEllipse
```

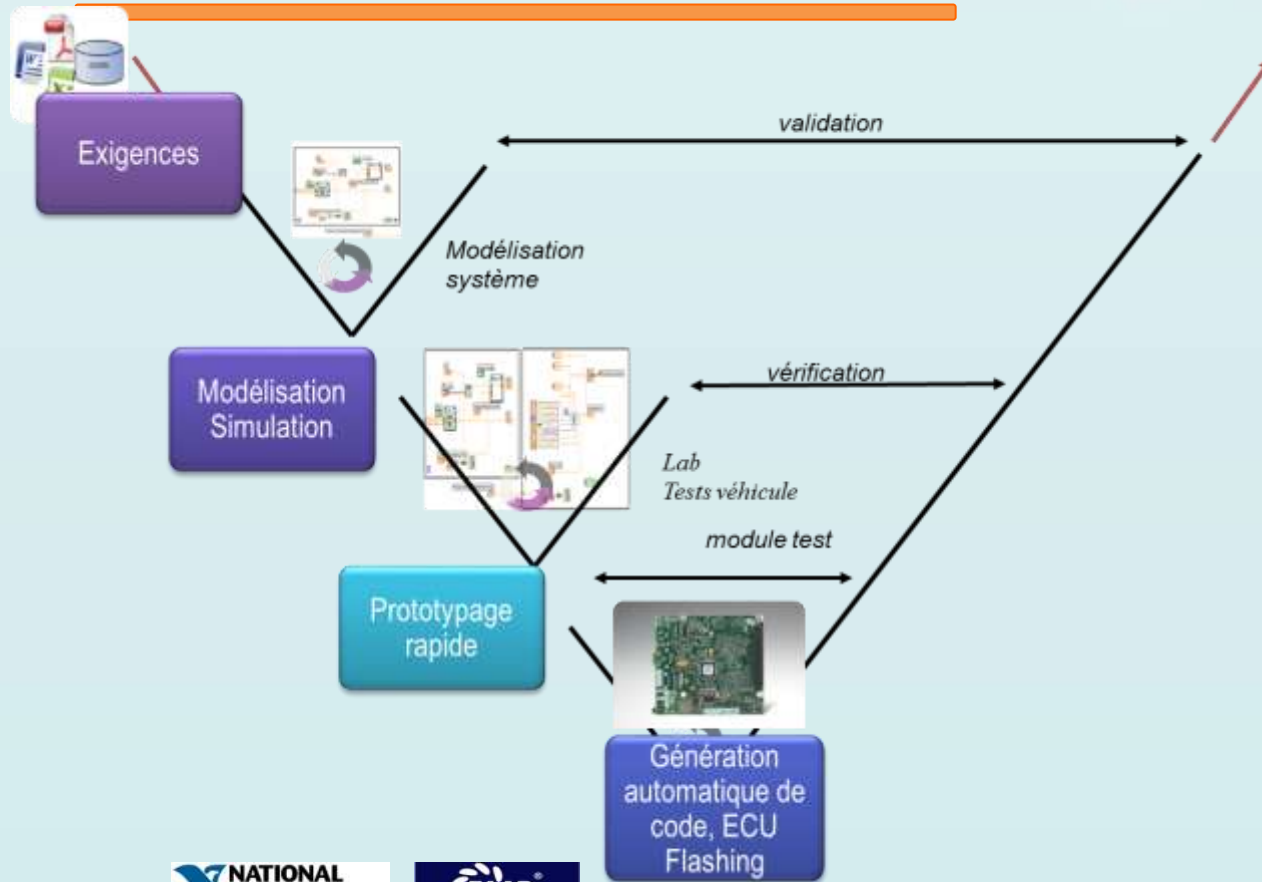


Optimiser vos temps de développement et la qualité de vos produits grâce au « model based design »

- 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



```
//gnc  
CreateRectPgn(80,160,80+  
gn.PtInRegion(point))  
  
void* x0c,y0c;  
for (int i=0;i<NUMCITY;i++)  
x0c=80+RectSizeX*(sEm)  
y0c=160+RectSizeY*(yEm)  
-gn.RgnEllipse;  
- CreateEllipse
```



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**

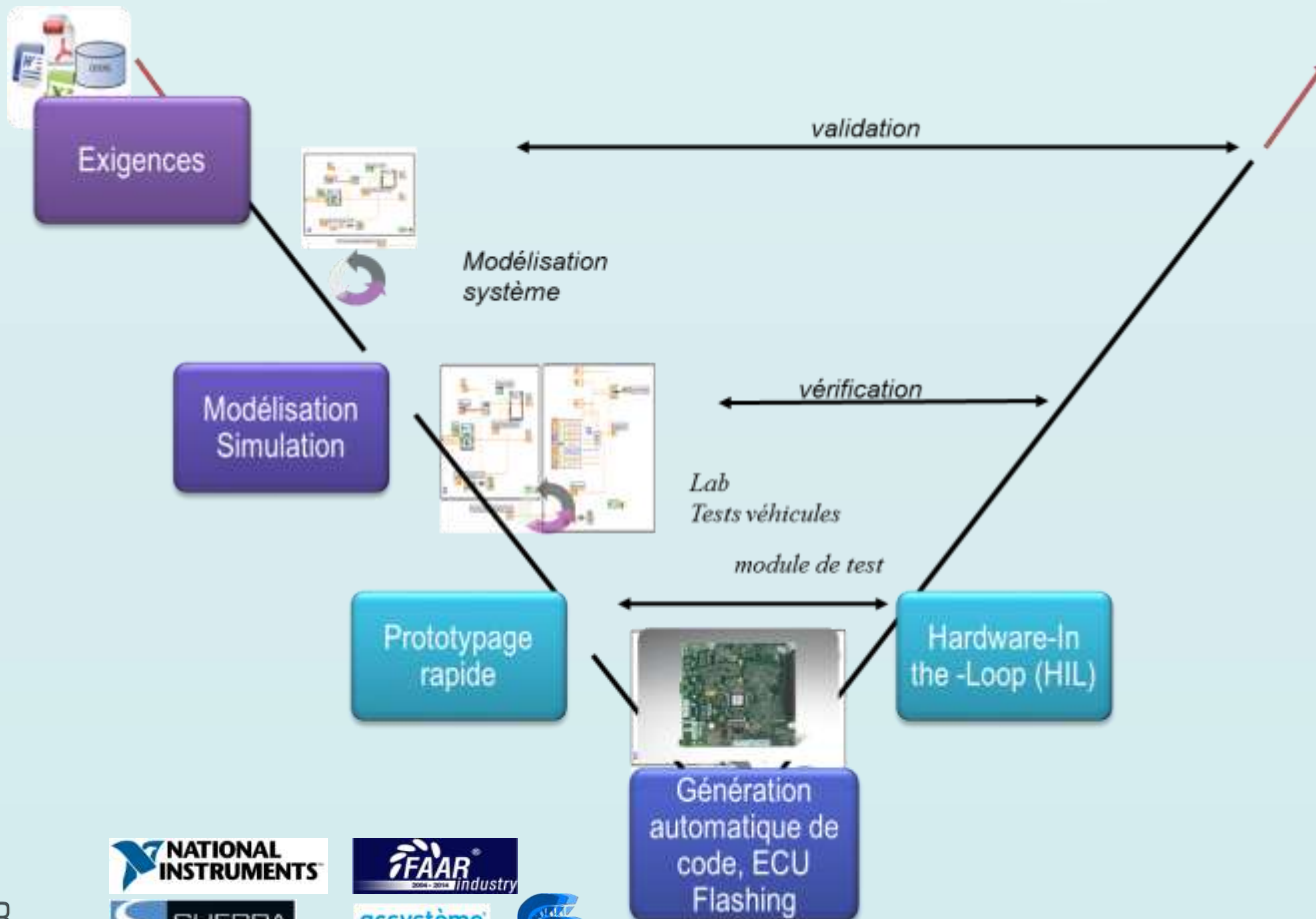



```

// Create Rect Pgm (80,160,80+
// Pgm.PtInRegion(point))
//
// module x0c,y0c;
// for (int i=0;i<NUNCITY;i++)
// {
//   x0c=80+RectSizeX*(i%NUNCITY);
//   y0c=160+RectSizeY*(i/NUNCITY);
//   Pgm.RectPgm(x0c,y0c,RectSizeX,RectSizeY);
//   Pgm.PtInRegion(point);
// }

```

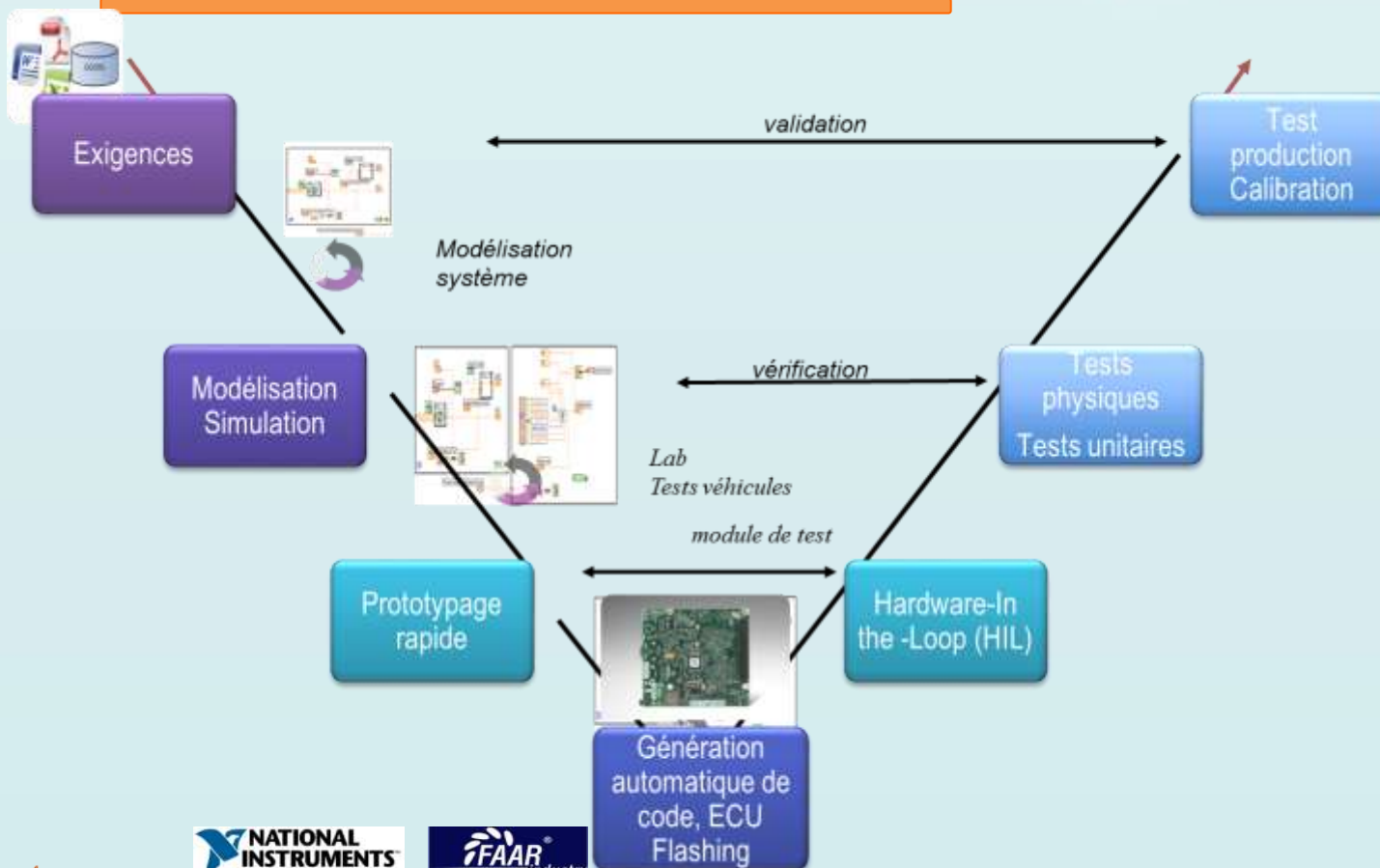
Validation HIL




```

// Create Rect Pgm (80, 160, 80+
// gn. Pt In Region (point))
//
// module x0c, y0c;
// for (int i=0; i<NUMCITY; i++)
// {
//     x0c=80+RectSizeX*(sExit
//     y0c=160+RectSizeY*(yExi
//     rgnEllipse:
//     rgnEllipse: CreateEllipse
  
```

Validation du produit fini



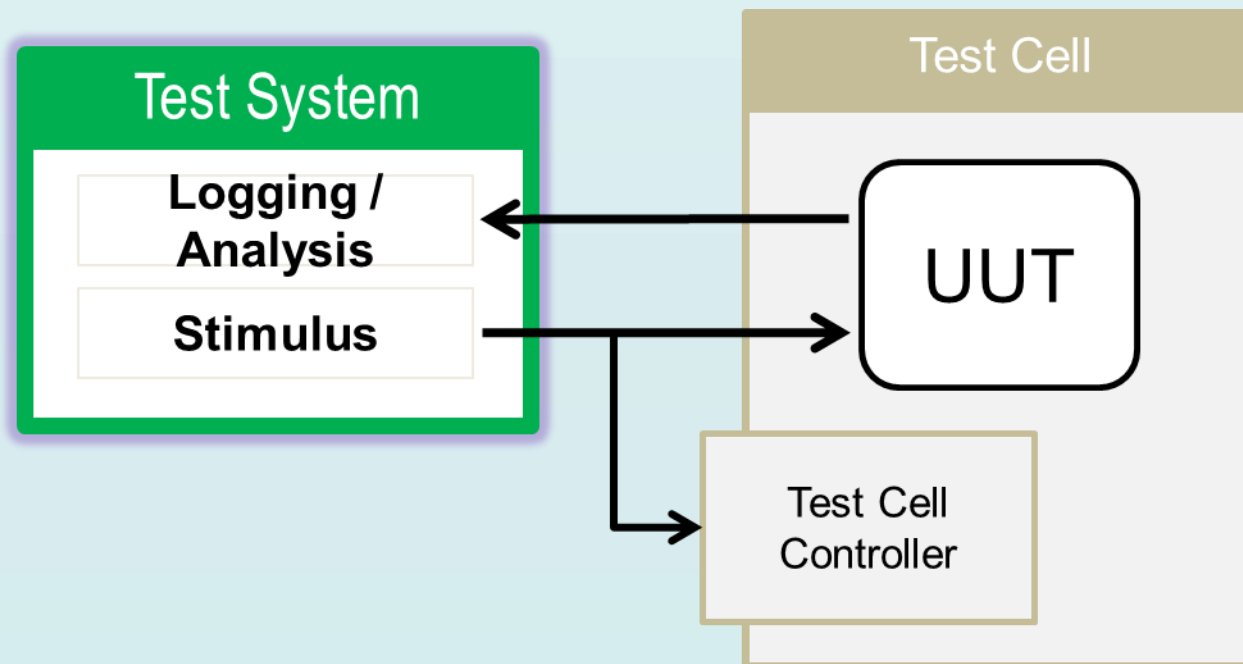
```

gfn:
CreateRectPgn(80,160,80+
gn.PtInRegion(point))
module xOc,yOc:
for (int i=0;i<NUMCITY;i++)
xOc=80+RectSizeX*(sExit
yOc=160+RectSizeY*(yExi
var rgnEllipse:
CreateEllipse

```



Validation du produit fini



```

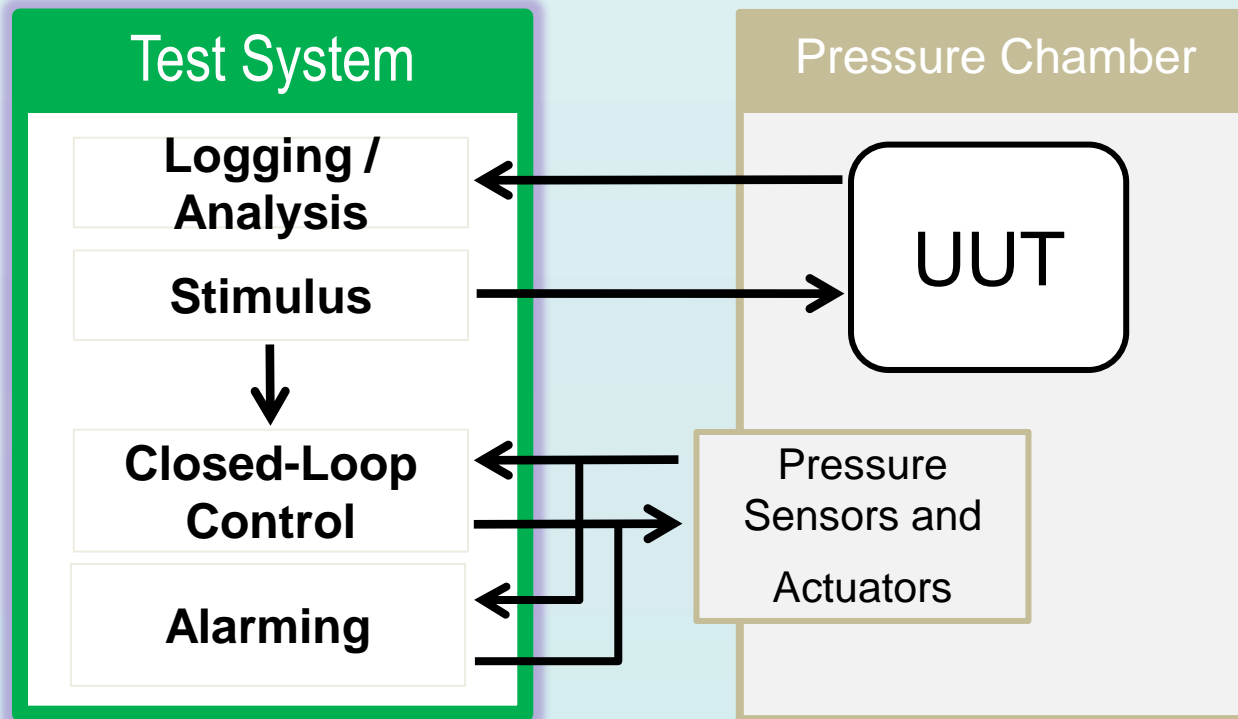
CreateRectPgn(80,160,80+
gn.PtInRegion(point))

void* xOc,yOc;
for (int i=0;i<NUMCITY;i++)
{
    xOc=80+RectSizeX*(sEmi
    yOc=160+RectSizeY*(yEmi
    rgnEllipse= CreateEllipse

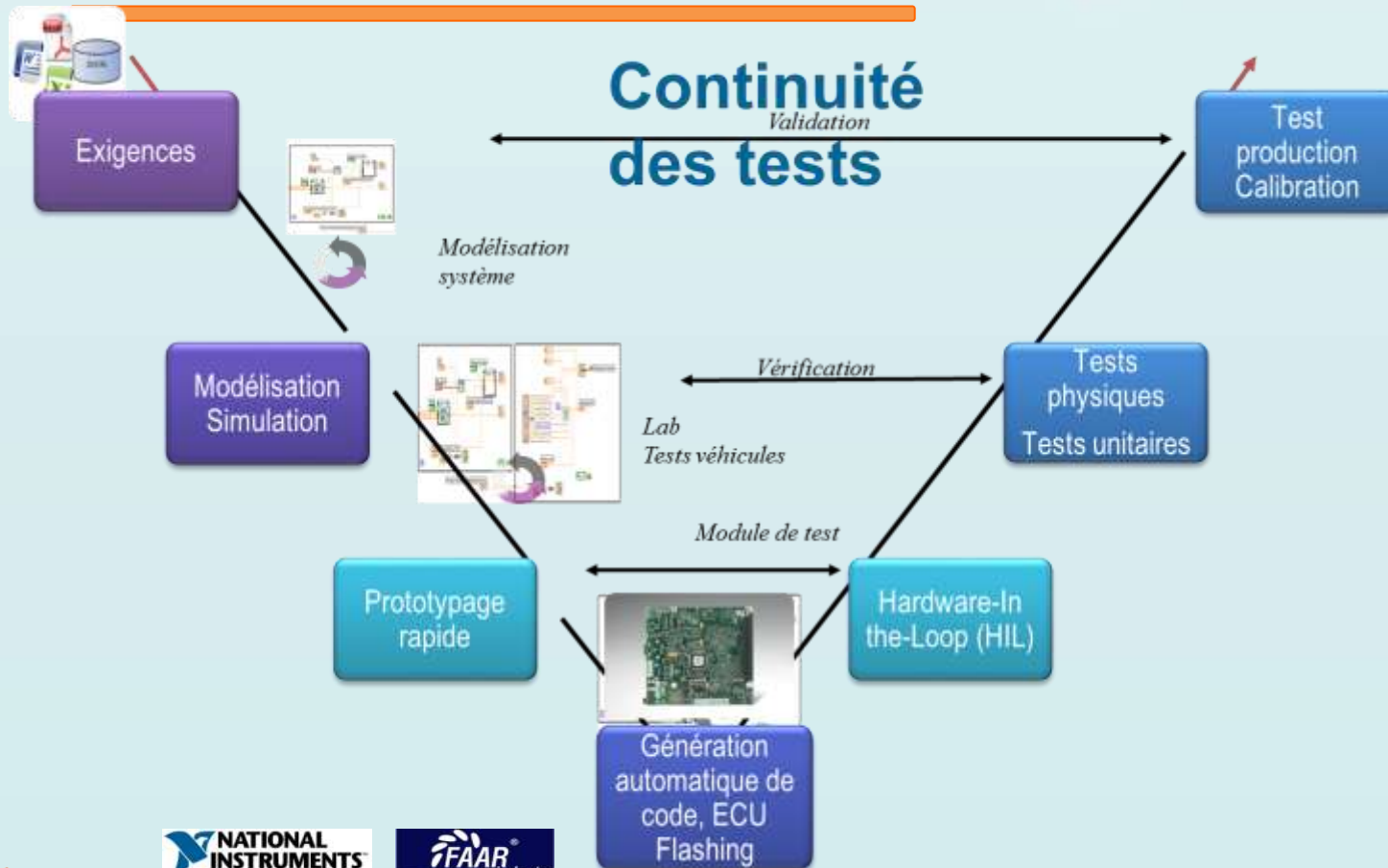
```



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 :

