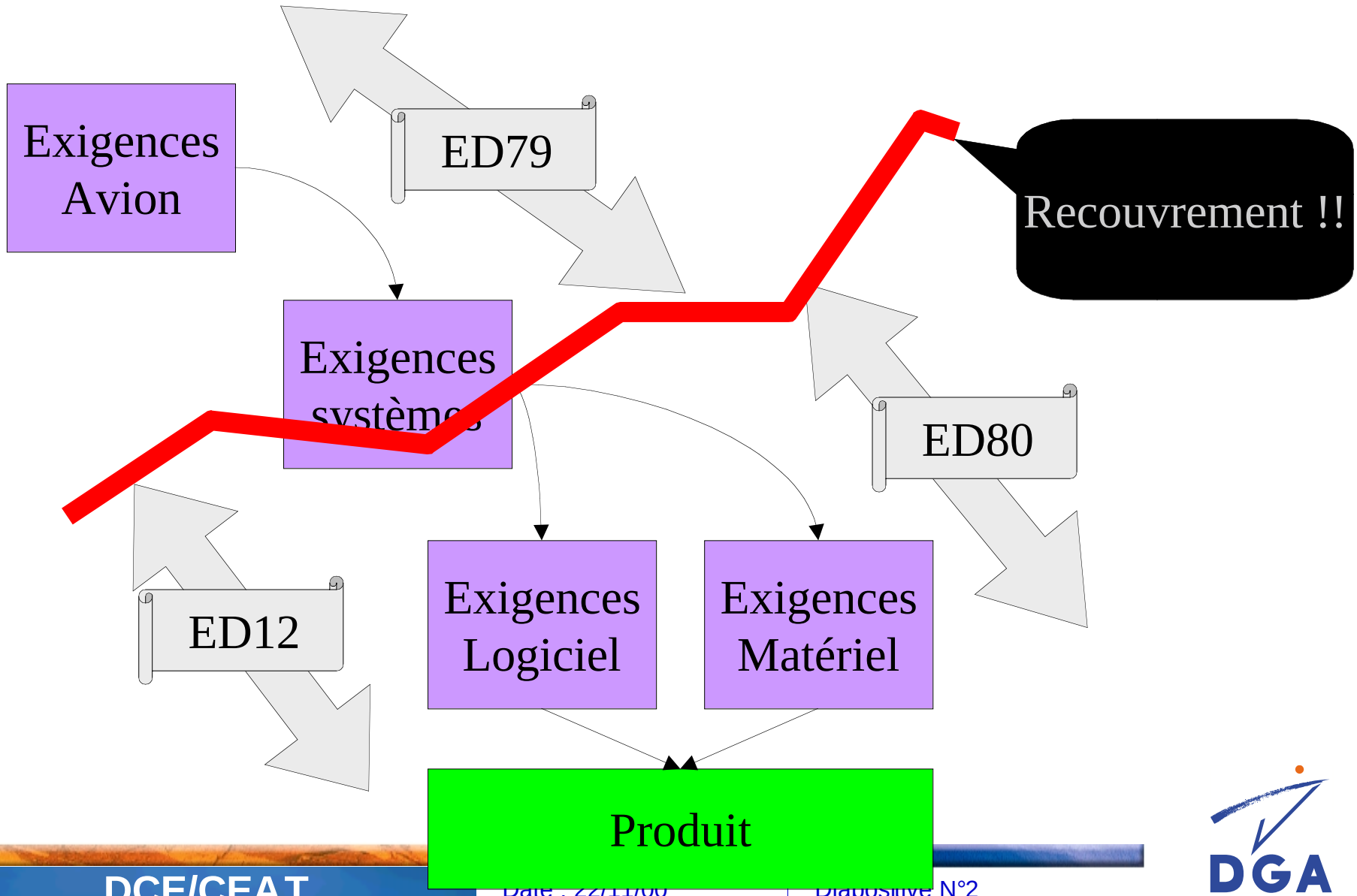
Logiciels et Certification

- Le logiciel n ’existe pas
- Textes réglementaires aéronautiques
- Moyens de conformité
- Expérience

Le logiciel n'existe pas



Les textes réglementaires

JAR 25.1301

– *Each item of installed equipment must*

- *(a) Be of a kind and design appropriate to its intended function;*
- *(b) Be labelled as to its identification, function, or operating limitations, or any applicable combination of these factors. (See ACJ 25.1301(b).)*
- *(c) Be installed according to limitations specified for that equipment; and*
- *(d) Function properly when installed.*

Les textes réglementaires (suite)

JAR 25.1309

– …systems…must be designed to ensure that they perform their intended functions under any foreseeable operating conditions.

Les moyens de conformité

Eurocae ED79, ED12B, ED80

Les processus

- Planification
- Développement
- Vérification
- Configuration
- Assurance Qualité

Les objectifs sont ceux de la JAR

Comment s'en sortir ?

- COTS : expérience
- Reverse Engineering
  - Permet le développement prototypal
- Certification incrémentale
  - Portabilité
  - Modularité (réutilisation)

Logiciel libre

- Quelle approche prendre ?
- Comment convaincre (les autorités) ?
- Est-ce moins cher in fine ?

COTS ?

- Pas de documentation (au sens ED12)
  - voire pas de code source
- Pas de droit de modification
  - propriété intellectuelle
- Logique de mise à jour commerciale

↪ PDS : Previously Developed Software

Reverse engineering ?

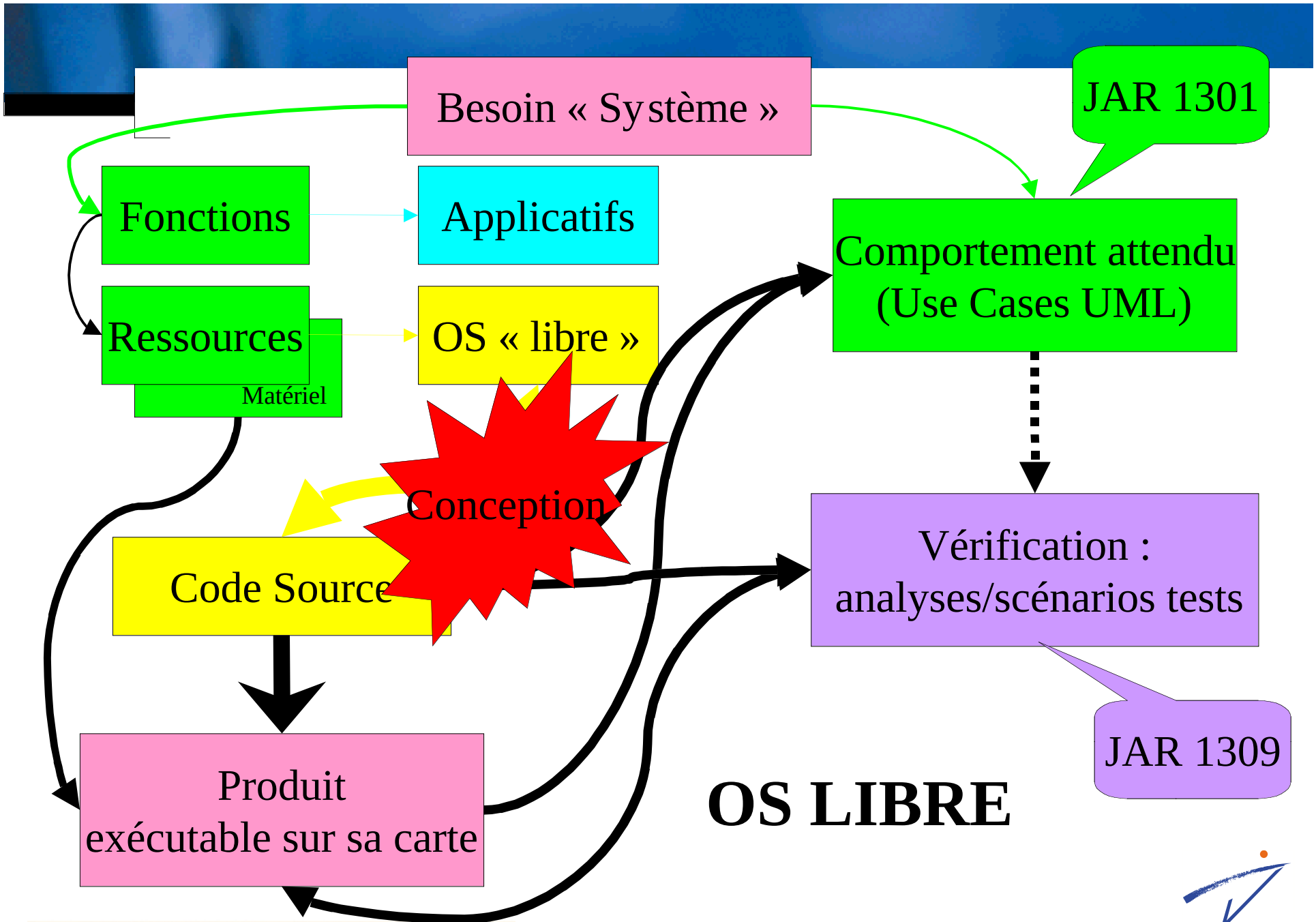
- Raccord entre le niveau système et le niveau logiciel
  - traçabilité
- Exigences trop détaillées
- Tests « structuraux »
- Difficulté de gestion du projet
  - absence d ’indicateurs

La certification incrémentale ?

- Portabilité
  - Comment rendre le fonctionnement « indépendant » de la plate-forme ?
- Modularité (réutilisation)
  - Intégration : problèmes aux interfaces
  - Modes communs

Exemples d ’application

- OS, librairies, drivers, protocoles
- Peu dans l ’embarqué
  - qques exemple, niveaux A à D.
- « sols »
  - maintenance (HUMS)
  - communication


Besoin « Système »
JAR 1301
Fonctions
Applicatifs
Comportement attendu
(Use Cases UML)
Ressources
Matériel
OS « libre »
Conception
Code Source
Vérification :
analyses/scénarios tests
JAR 1309
Produit
exécutable sur sa carte
OS LIBRE

Difficultés

- Manquent
  - Les exigences de bas niveau
  - les aspects vérification du flot de contrôle et de données
- ex séquenceur : comment vérifier le comportement au limite ? (de la machine à états)

Conclusion

- Révolution Culturelle
  - renouvellement des générations
  - « Jeter » les standards
- Introduction contrôlée
  - faible niveau (D ou C)
- Attention à l'accoutumance !
- Fonctions de maintenance