

Quelle organisation pour développer ?



Les principes et les valeurs de
l'extreme programming **XP** sont bons

Quelle organisation pour développer ?



Principes

- Revue de code
- Production systématique de cas tests
- *Refactoring*
- Solutions simples
- Métaphores
- Intégration quotidienne
- cycles de développement très rapides

[http://fr.wikipedia.org/wiki/Extreme_programming]

Quelle organisation pour développer ?



Valeurs

- La communication
- La simplicité
- Le feedback
- Le courage
- Le respect

[http://fr.wikipedia.org/wiki/Extreme_programming]

Toute chose appartient à qui la rend meilleure

Debugging is fun because it begins with a mystery



Les méthodes agiles **SCRUM** sont bonnes

- Individus et interactions vs. Processus et outils
- Logiciel qui fonctionne vs. Documentation exhaustive
- Collaboration du client vs. Négociation de contrat
- Réponse au changement vs. Suivi d'un plan prédéfini

[<http://fr.wikipedia.org/wiki/Scrum>]

Quelle organisation pour développer ?



Courage et Solidarité

Appropriation collective du code

L'équipe est collectivement responsable de l'application. Chaque développeur peut faire des modifications dans toutes les portions du code, même celles qu'il n'a pas écrites. Les tests diront si quelque chose ne fonctionne plus.

Refactoring (ou remaniement du code)

Amélioration régulière de la qualité du code sans en modifier le comportement. On retravaille le code pour repartir sur de meilleures bases tout en gardant les mêmes fonctionnalités. Les phases de *refactoring* n'apportent rien au client mais permettent aux développeurs d'avancer dans de meilleures conditions, et donc plus vite.

[http://fr.wikipedia.org/wiki/Extreme_programming]

Quelle organisation pour développer ?



Langage commun et communication fluide

Convention de nommage

Puisque tous les développeurs interviennent sur tout le code, il est indispensable d'établir et de respecter des normes de nommage pour les variables, routines, méthodes, objets, classes, fichiers, etc.

Utilisation de métaphores

On utilise des métaphores et des analogies pour décrire le système et son fonctionnement. Le fonctionnel et le technique se comprennent beaucoup mieux lorsqu'ils sont d'accord sur les termes qu'ils emploient.

[http://fr.wikipedia.org/wiki/Extreme_programming]

Quelle organisation pour développer ?



Qualité, utilisabilité, réactivité

Intégration continue

Lorsqu'une tâche est terminée, les modifications sont immédiatement intégrées dans le produit complet. On évite ainsi la surcharge de travail liée à l'intégration de tous les éléments avant la livraison. Les tests facilitent grandement cette intégration : quand tous les tests passent, l'intégration est terminée.

Tests unitaires

Avant d'implémenter une fonctionnalité, le développeur écrit un test qui vérifiera que son programme se comporte comme prévu. Ce test sera conservé jusqu'à la fin du projet, tant que la fonctionnalité est requise. À chaque modification du code, on lance tous les tests écrits par tous les développeurs, et on sait immédiatement si quelque chose ne fonctionne plus.

[http://fr.wikipedia.org/wiki/Extreme_programming]

Quelle organisation pour développer ?



Un exemple d'atelier de génie logiciel :
Code_Aster

Modèle collectif ouvert, de type logiciel libre

Quelle organisation pour développer ?



1. Identifier le périmètre de responsabilité
2. Contractualiser le versionnement
3. Lotir et organiser les travaux
4. Communiquer et intégrer

Quelle organisation pour développer ?



Identifier le périmètre de responsabilité : La fiche qualité / le plan qualité logiciel

2 Désignation de la version qualifiée

La version 7 qualifiée du *Code_Aster* sur plate-forme AlphaServer TRU64 et Linux est composée de :

Logiciel	Version	Date de l'exécutable	Machine d'exécution
Code_Aster			
• Solveur Aster	7.4	08/12/2004	TRU64, Linux
• askt	1.2	07/01/2005	TRU64, SUN, Linux
• eficas	1.7	08/12/2004	TRU64, SUN, Linux
• agla	5.22	06/04/2005	TRU64

Tableau 2-1

3 Environnement informatique de validation du solveur Aster

Type de machine	AlphaServer ES45
Version du système d'exploitation	TRU64 OSF1 v5.1A
Version du compilateur C	CC 6.5
Version du compilateur Fortran	F90 5.5
Répertoire et support de stockage	précisé en [A1.01.10]

Tableau 3-1

Type de machine	PC Linux (32 bits)
Version du système d'exploitation	Calibre (distribution EDF) ou autres Linux (cf. remarque ci-dessous)
Version du compilateur C	gcc 3.3
Version du compilateur Fortran	g77 3.3

Tableau 3-2

6 Liste des fonctionnalités non qualifiées

Certaines modélisations, ne satisfaisant pas aux exigences qualité, sont exclues des domaines validés pour la version 7 en exploitation et ne doivent donc pas être utilisées pour des études sous AQ. Ces modélisations concernent les lois de comportement suivantes :

2 Liste et version des outils

Produit	Version	Description
<i>Les logiciels suivants sont indisponibles</i>		
Python	2.2.2	Langage de programmation de haut niveau
tk/tk	8.4	Kit de développement graphique
eficas	1.7	Logiciel de calcul de la déformabilité
Pmw	0.8.5 ou 1.2	Python widget pour encastrés
numerical python	20.3	Manipulation de tableaux en python
med	2.2.2	Bibliothèque d'échange de données
hdf5	1.4	Bibliothèque de compression pour med
askt	1.2	Interface de lancement (inclus bsf, asjob, arex)
metis	metis-edf-4.0	Outil de renumérotation (onmetis)
<i>Si les logiciels suivants sont absents, seules quelques opérateurs du code sont indisponibles</i>		
xmgrace	5.1	Traceur de courbes (pour STANLEY, IMPR_TABLE, IMPR_FONCTION)
homard	7	Logiciel de calcul de la déformabilité
mefisto	3	Logiciel de calcul de la déformabilité
miss3d	1	Logiciel de calcul de la déformabilité
gibi	2000	Maillage et post-traitement (pour MACR_ASCOU/FASPIC_MAIL)
gmsb	1.60	Maillage et post-traitement (pour STANLEY)
gnuplot	3.7.1	Traceur de courbes (pour MACR_RECAL)
gnuplot.py	1.5	Module python interface avec gnuplot
<i>Les logiciels suivants sont des outils périphériques, non appelés directement par les commandes du code</i>		
agla	5.22	Atelier de Génie Logiciel
agraf	3	Logiciel de calcul de la déformabilité
i-deas	7	Logiciel de calcul de la déformabilité
ensight	7	Logiciel de calcul de la déformabilité
acrobat reader	5.0	Visualiseur de fichier PDF
navigateur internet		Consultation du site www.code-aster.org

Logiciels indispensables pour utiliser Aster

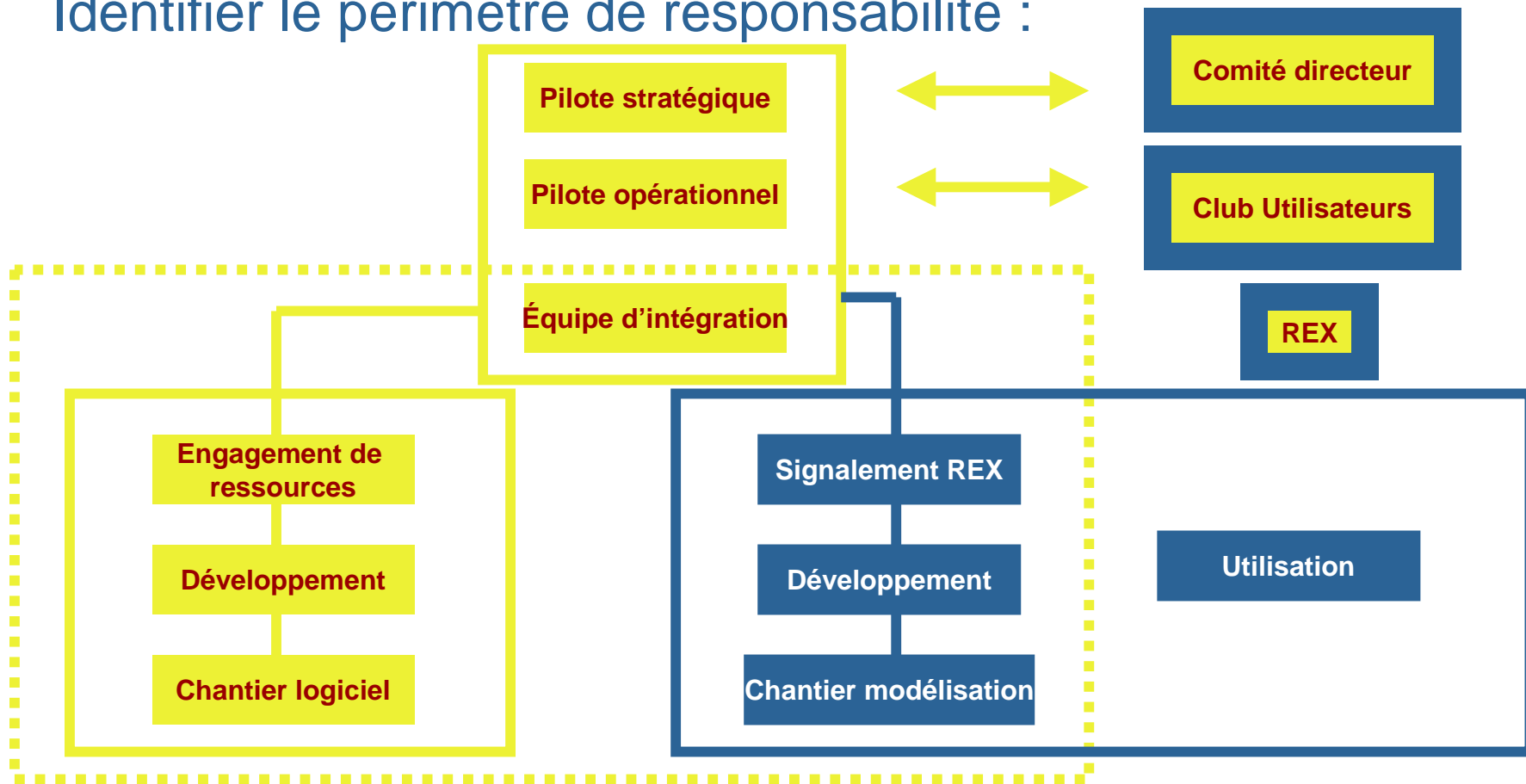
Logiciels nécessaires à certaines commandes

Outils périphériques

Quelle organisation pour développer ?



Identifier le périmètre de responsabilité :



Équipe Coeur

Utilisateurs et chantiers applicatifs



Quelle organisation pour développer ?



Gestion de configuration :

- Pas de cycle en V à l'échelle du code **XP**
- Releases hebdomadaires **XP**
- Versions de développement utilisables et accessibles **XP**
- Mise en exploitation régulière de versions qualifiées

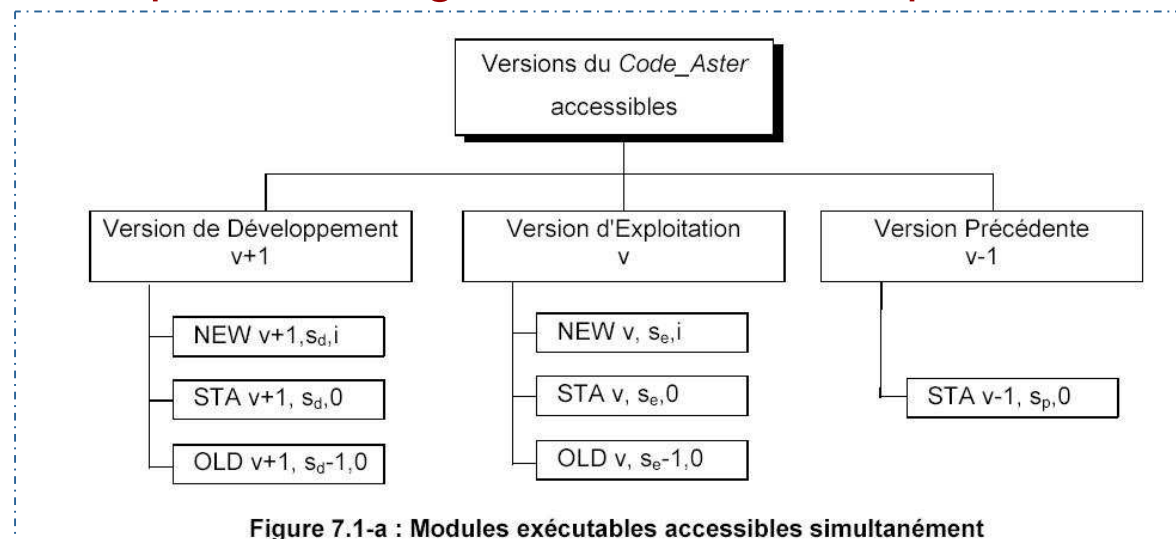
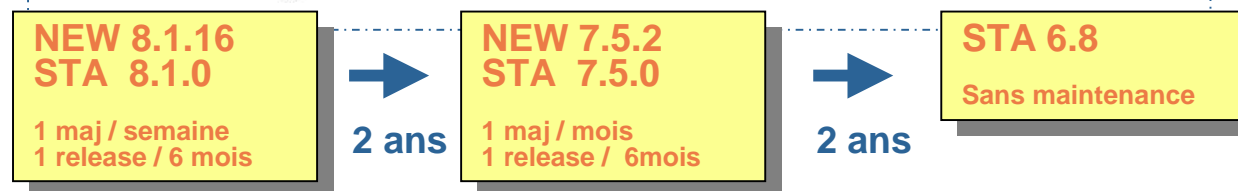


Figure 7.1-a : Modules exécutables accessibles simultanément



Quelle organisation pour développer ?



Lotir les travaux : types d'interventions sur le code :

Nature	Origine	Canal
Cas test « cassé »	Développeur	Mise à jour
Signalement de dysfonctionnement	Utilisateur	REX
Lotissement chantier applicatif	Développeur / Utilisateur	REX
Évolution noyau	Développeur / Utilisateur	REX + planification activité
Requête utilisateur	Utilisateur	REX
Refactoring	Développeur	Initiative collective et individuelle



Quelle organisation pour développer ?



Lotir les travaux

Découper chaque chantier en interventions autonomes et restituables :

- De plus petite dimension possible
- Qui feront l'objet de fiches REX individuelles
- Qui seront utilisables, documentables, testables
- Qui seront intégrées dans le code sans délai

XP + SCRUM, méthodes agiles



Quelle organisation pour développer ?



Organiser le développement : outil de REX

Le REX est plus qu'un bug-tracker :

Outil de gestion de tous types d'intervention sur le code

Code_Aster REX Aster

issue 11748 nosy, title edited ok

Edition Fiche 11748

mes requetes <ul style="list-style-type: none">AL enregistréesAOMEL enregistréesEfficasTMA CSTMA Deltacadà compléter	Titre X-FEM : définition de fissures et orientation automatique du front	Etat actuel ferme	Nouvel etat ferme
fiches <ul style="list-style-type: none">NouvelleEmisesEnregistréesToutesRecherche <p>Voir fiche: <input type="text"/></p>	Type evolution		
administration <ul style="list-style-type: none">Liste des classesUtilisateursNouvel utilisateur	Produit Code_Aster		
bonjour, durand christophe <ul style="list-style-type: none">Mes infosMes fichesFiches à corrigerSe déconnecter	Diffusion bugmaster,courtois,galenne,geniaut (list)		
aide <ul style="list-style-type: none">Roundup docs	Emetteur GALENNE Erwan	Organisation EDF AMA/T64	
<small>Dans version 9.3.6.0.0</small>	Projet ANODE	RTA <input checked="" type="checkbox"/> Validé <input type="checkbox"/> Non valide	Intervenant
	Version cible 9.4		
	Responsable LAVERNE Jérôme		
	Type de Validation sslv110f, sslv134h, sslv313b, ssnv13		
	Version d'exploitation:		
	Corrigée <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	Faux depuis version	<input type="text"/>
	Corrigé dans version		
	Version de développement:		
	Corrigée <input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Faux depuis version	<input type="text"/>
	Corrigé dans version 9.3.6		
	Impact documentation U4.82.08, U4.82.21, U4.82.22, U4.82.23	Nombre de Jours	<input type="text" value="7.0"/>



Quelle organisation pour développer ?



Organiser le développement : outil de REX

Code_Aster

REX Aster

issue 11748 nosy, title edited
ok

Edition Fiche 11748

mes requetes

- AL enregistrées
- AOM
- EL enregistrées
- Eficas
- TMA CS
- TMA Deltacad
- à compléter

fiches

- Nouvelle
- Emises
- Enregistrées
- Toutes
- Recherche

Voir fiche:

administration

- Liste des classes
- Utilisateurs
- Nouvel utilisateur

bonjour, durant christophe

- Mes infos
- Mes fiches
- Fiches à corriger
- Se déconnecter

aide

- Roundup docs

Titre	X-FEM : définition de fissures et orientation automatique du front	Etat actuel	ferme
Type	evolution	Nouvel etat	ferme
Produit	Code_Aster		
Diffusion	bugmaster courtrois.galenne.geniaut (list)		
Emetteur	GAL		
		MA/T64	
		RTA	<input checked="" type="radio"/> Validé <input type="radio"/> Non validé
Responsable	LAVERNE Jérôme	Intervenant	
Type de Validation	sslv110f, sslv134h, sslv313b, ssnv18		
Version d'exploitation:	Corrigée <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
Version de développement:	Corrigée <input checked="" type="checkbox"/> Yes <input type="checkbox"/> No		
Impact documentation	U4.82.08, U4.82.21, U4.82.22, U4.82.23		
		de	7.0
		Jours	

Identifiant de l'intervention sur le code
Tracé dans les pv de mise à jour

Avancement de la fiche :

- Émise
- Enregistrée
- Résolu
- Validé_EDA
- Fermé

Quelle organisation pour développer ?



Organiser le développement : outil de REX

Code_Aster REX Aster

issue 11748 nosy, title edited ok

mes requetes

- AL enregistrées
- AOM
- EL enregistrées
- Eficas
- TMA CS
- TMA Deltacad
- à compléter

fiches

- Nouvelle
- Emises
- Enregistrées
- Toutes
- Recherche

Voir fiche:

administration

- Liste des classes
- Utilisateurs
- Nouvel utilisateur

bonjour, durant christophe

- Mes infos
- Mes fiches
- Fiches à corriger
- Se déconnecter

aide

- Roundup docs

Émetteur de la demande

Titre: X-FEM : définition d
Type: evolution
Produit: Code_Aster
Diffusion: bugmaster,courtois,galenne,geniaut,(list)
Émetteur: GALENNE Erwan
Etat actuel: ferme
Nouvel état: ferme

Contexte projet

Projet: ANODE
Version cible: 3.4
Responsable: LAVERNE Jérôme
RTA: Validé Non validé
Intervenant

Responsable du travail à réaliser

Type de Validation: sslv110f, sslv134h, sslv133b, ssnv18
Version d'exploitation:
Corrigée: Yes No
Version de développement:
Corrigée: Yes No
Faux depuis version:
Corrigé dans version: 9.3.6
Impact documentation: U4.82.08, U4.82.21, U4.82.22, U4.82.23
Nombre de Jours: 7.0

Rev version 03/06/08

Quelle organisation pour développer ?



Organiser le développement : outil de REX

Code_Aster

REX Aster

issue 11748 nosy, title edited
ok

Edition Fiche 11748

mes requetes

- AL enregistrées
- AOM
- EL enregistrées
- Eficas
- TMA CS
- TMA Deltacad
- à compléter

fiches

- Nouvelle
- Emises
- Enregistrées
- Toutes
- Recherche

Voir fiche:

administration

- Liste des classes
- Utilisateurs
- Nouvel utilisateur

bonjour, durant christophe

- Mes infos
- Mes fiches
- Fiches à corriger
- Se déconnecter

aide

- Roundup docs

Validation, cas tests

Tracé qualité

Impact sur versions

Impact documentation

Impact

Titre: X-FEM : définition de fissures et orientation automatique du front

Etat actuel: ferme

Nouvel état: ferme

Organisation:

Projet: ANODE

Version cible: 9.4

Responsable: LAVERNE Kévin

Type de Validation: sslv110f, sslv134h, sslv313b, ssnv18

Version d'exploitation:

Corrigée: Yes No

Version de développement:

Corrigée: Yes No

Faux depuis version:

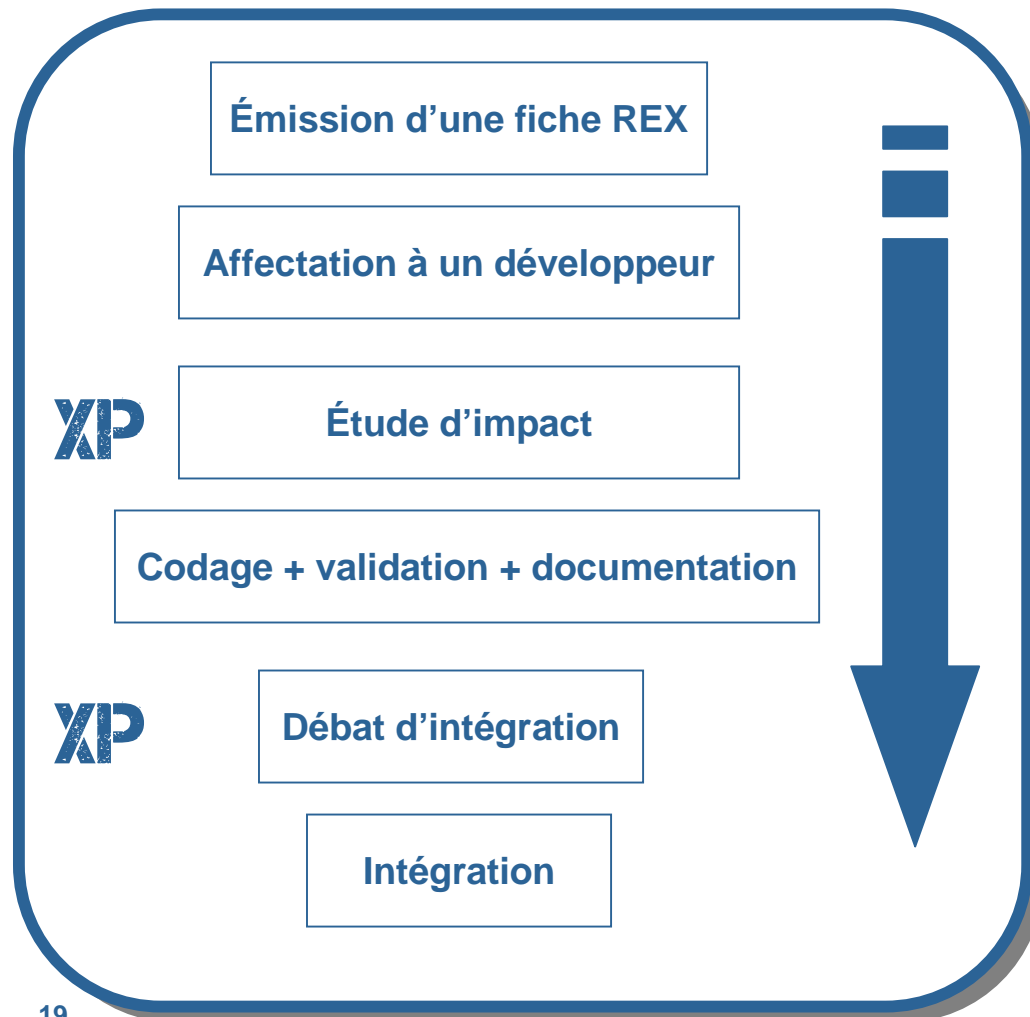
Impact documentation: U4.82.08, U4.82.21, U4.82.22, U4.82.23

Nombre de Jours: 7.0

Quelle organisation pour développer ?



Le cycle de vie d'une intervention sur le code :



Affectation hebdomadaire
par équipe noyau

Tous acteurs concernés

Obligatoire.
Réunion hebdomadaire de l'équipe noyau
+ tous développeurs

Environnement de gestion de sources
+ contrôle qualité des développements



Quelle organisation pour développer ?



Importance de la communication et des métaphores :



Créer des **structures** (réunions, comités ...) officielles de contrôle des développements :

- En amont (pré-développement) :

Pour préciser le besoin fonctionnel avec l'utilisateur

Pour réaliser l'étude d'impact architectural avec l'équipe noyau

Ne rien faire à l'insu des responsables des sources impactés

- En aval (pré-intégration) :

Fonction d'**exigence** : tests, documentations

Fonction de **vigilance** : robustesse, performances, utilisabilité

Fonction de **cohérence** : sources, interface utilisateur



Quelle organisation pour développer ?



Les trois paliers (exigences) du développement :

① Fiabilité :

Un modèle doit donner des résultats justes

② Robustesse :

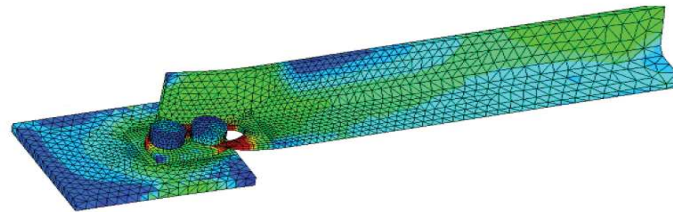
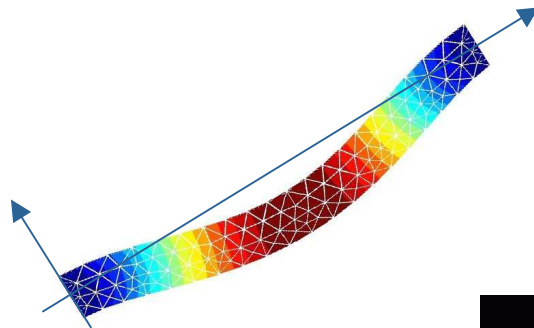
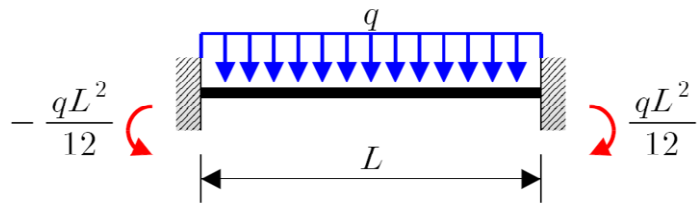
Un modèle quelconque (industriel) doit donner des résultats fiables ①

③ Performance :

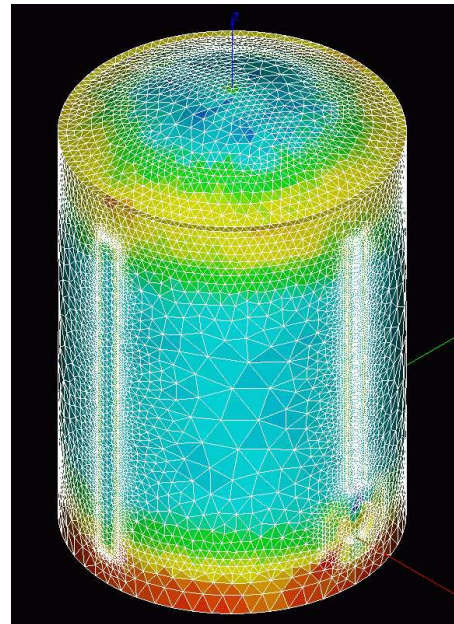
Un modèle quelconque (industriel) de grande taille doit donner des résultats fiables ① et robustes ②

Ne pas aborder la mise au point dans un autre ordre !

Quelle organisation pour développer ?



- ① Fiabilité
- ② Robustesse
- ③ Performance



Quelle organisation pour développer ?



Qui développe ? Dualité langage haut niveau / langage bas niveau

Fortran

Piliers du code – architectes d’opérateurs
Intégrateurs de nouveaux modèles

API bibliothèques
Intergiciel haut/bas niveau

Informaticiens purs

Python Fortran

Intégrateurs de modules pré-câblés (loi de cmpt)

Python
Macro-commandes

Développeurs application métier

Python
Fichier de commandes

Utilisateur

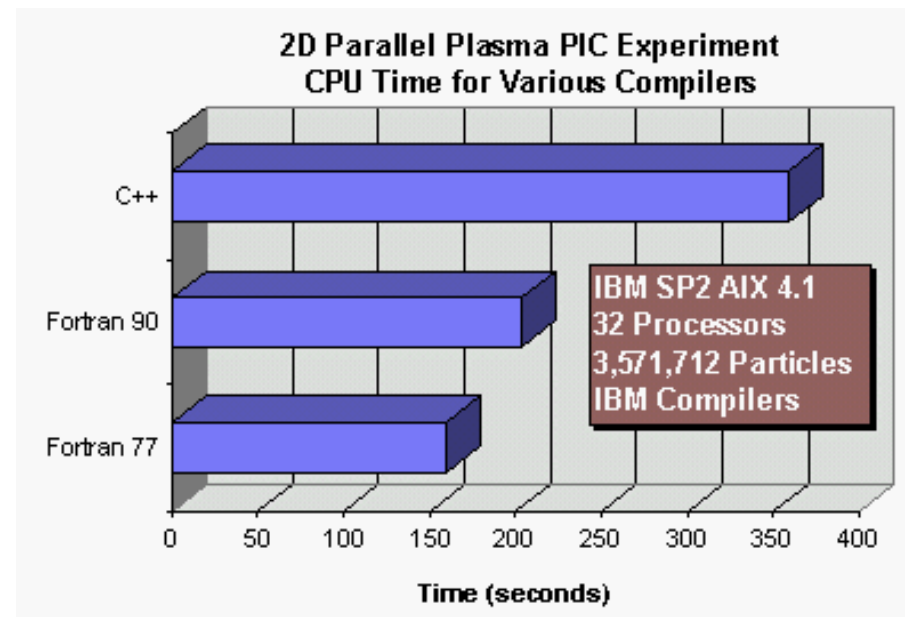
Quelle organisation pour développer ?



Choix de langage :

Haut niveau : **FORTRAN** pour performances et accessibilité pour tous

Bas niveau : **Python** pour puissance et accessibilité pour tous



<http://www.cs.rpi.edu/~szymansk/oof90.html>