open
engineering

Extreme Programming

Application
aux pratiques de développement et de gestion de projet

TOC

Spatiopole, rue des Chasseurs Ardennais, B-4031 Angleur, Belgium Tél. (+32)4/37.29.345 www.open-engineering.com

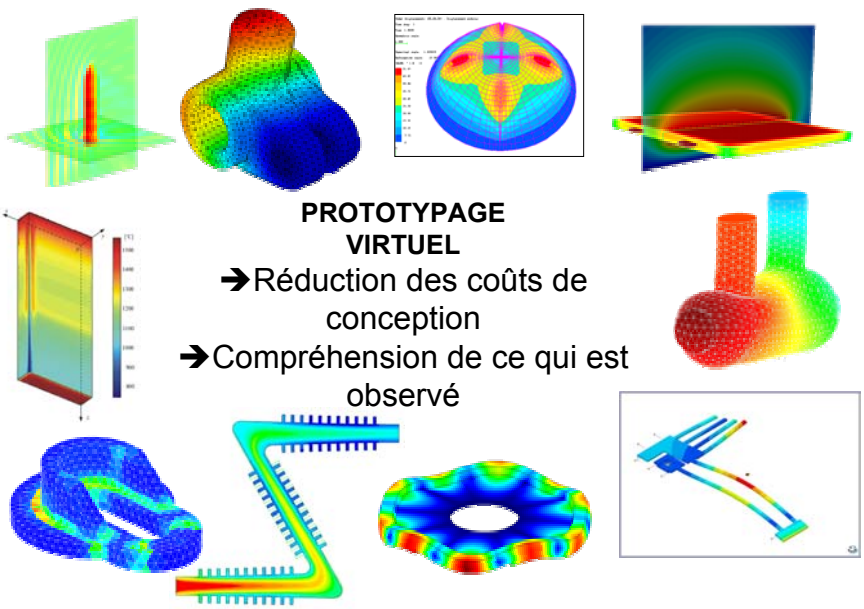
Open Engineering

Spin-off créée pour capitaliser 30 années de recherches menées à l'Ulg et l'Intec (Argentine)

Membre du groupe Samtech (http://www.samcef.com)

Objectif :

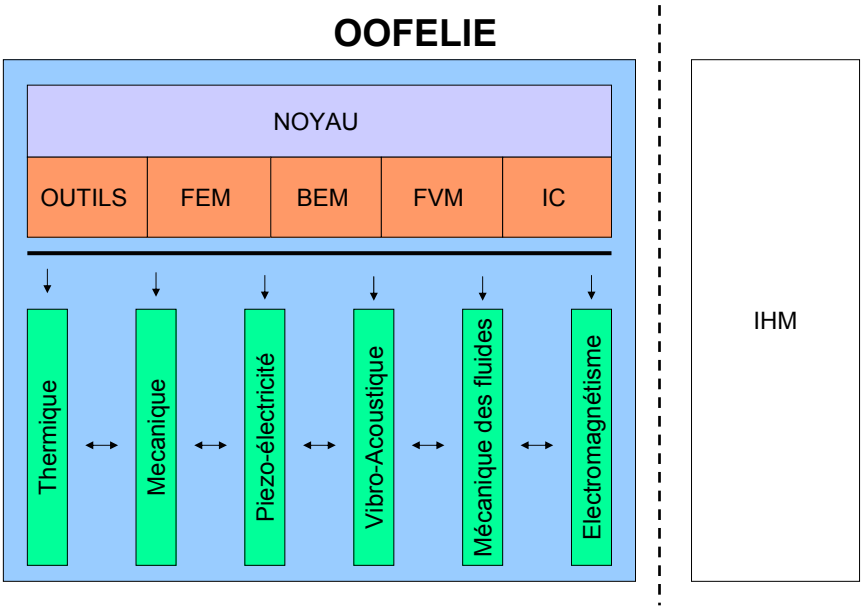
Bénéficier des avantages de l'adaptation des méthodes orienté objet à la simulation numérique pour résoudre des problèmes multi-disciplinaires en utilisant un toolkit numérique unifiant connu sous le nom de OOFELIE.



PROTOTYPAGE VIRTUEL

➔Réduction des coûts de conception

➔Compréhension de ce qui est observé



Les 13 pratiques de XP

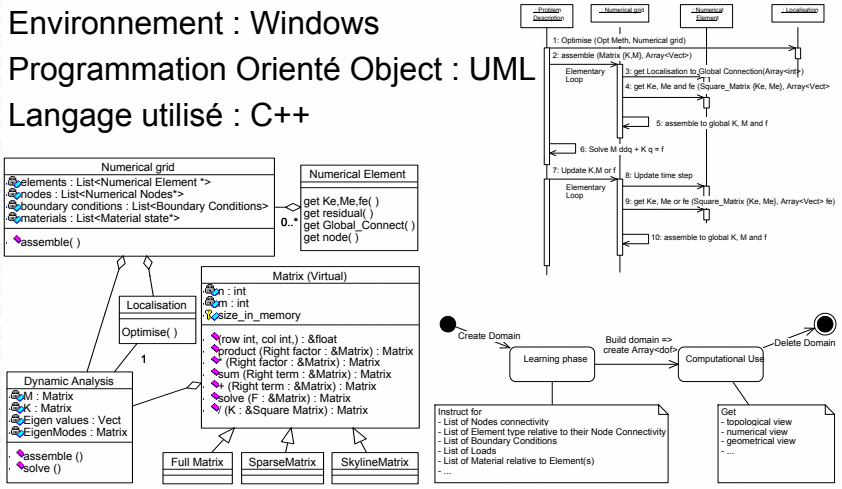
- Conception simple & refactoring
- Tests unitaires et de recettes
- Responsabilité collective du code
- Programmation en binôme
- Règles de codage
- Métaphore
- Intégration continue
- Livraisons fréquentes
- Planification itérative
- Client sur site
- Rythme durable

Environnement de développement

Environnement : Windows

Programmation Orienté Object : UML

Langage utilisé : C++



Conception simple & Refactoring

Simple ≠ Simpliste

But : code clair, lisible et compréhensible

Distinction anciens et nouveaux développements

Pourquoi? 10 années de co-développements “universitaires”

Impératif enrichir une batterie de tests

!!! Codeur « fou » (rend le code difficile à lire)

!!! Généralité spéculative (lourdeur inutile)

Références :

Livres : « Refactoring » (Addison Wesley)

« Design Patterns » (Addison Wesley)

Web : http://www.refactoring.com

Tests unitaires et de recettes

Automatiques (si non pas utilisé)

La batterie de tests contient

- Des tests unitaires (pour certaines classes)
- Des tests de recettes (validation fonctionnelle)

La batterie de tests sert à s'assurer

- Des non régressions (↗ la confiance de l'équipe)
- Afficher des capacités de résolution de problèmes (↗ la confiance des prospects)
- **Portage**

Importance de la relation avec le client

!!! La couverture des tests (déresponsabilisation)

Présentation de l'équipe

7,5 ingénieurs civil (dont 3 docteurs en sciences appliquées)

Compétences scientifiques complémentaires

Environnement de développement uniforme

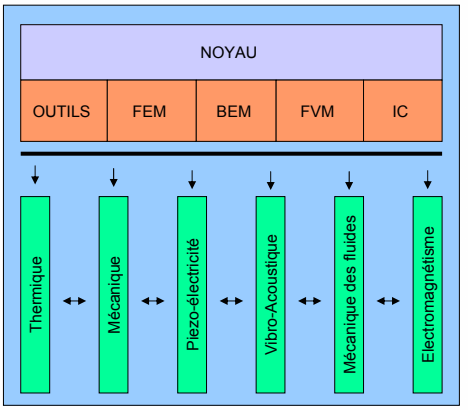
Communication

Responsabilité collective du code
Programmation en binôme

Partielle

Avantages observés :

- Partage des connaissances
- Gain de temps en formation
- Motivation
- Satisfaction



Règles de codage

Indispensable

Inspirée de l'Ellemtel mais adaptée à nos besoins

Adaptation collective

Distinction anciens et nouveaux développements

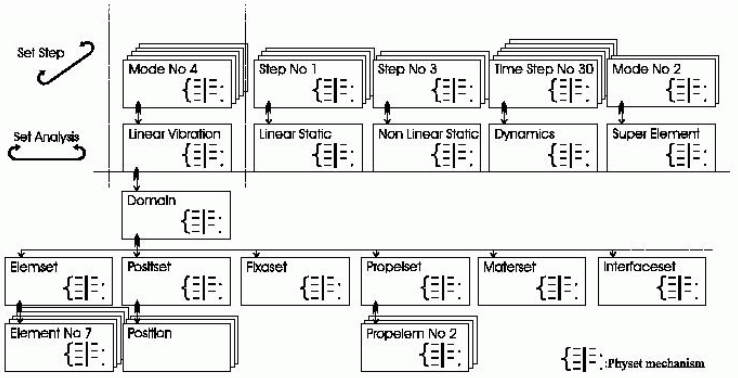
Référence :

Web : http://www.doc.ic.ac.uk/lab/cplus/c++.rules

Métaphore

Pas de globale

Existe local : ex Arbre PhySet



Intégration continue

Source sous serveur CVS ainsi que la batterie de tests et ses références

Mises en commun journalières

Compilation & test batterie « **nocturne** » multi-plateforme

Pour des raisons de convivialité, les développements se font dans l'environnement MS Windows (wincvs + MS développeur studio + cygwin).

La validation du portage est réalisée de manière automatique

Caractéristiques importantes :

- **Pas de mécanisme de réservation (« Locking »)**
- **Mécanisme de branches**

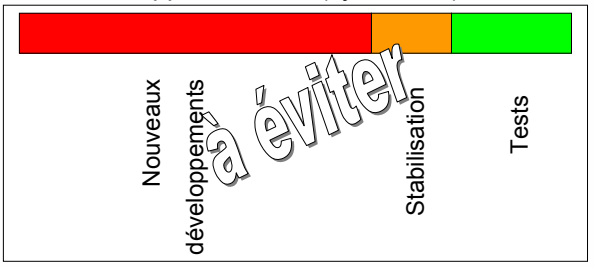
Livraisons fréquentes

Réalisable pour OOFELIE

Limitation pour nos produits à cause de l'IHM.

Causes?

1. Cycle de développement IHM (cycle en V)



2. Outil de versionnement (locking et pas de branches)

Planification itérative

Limitations de la planification en « V »

Déjà réalisée pour des projets externes

Importance élevée des tests fournis par le client **avant** les itérations afin d'anticiper les problèmes

Fait partie de tous nos projets.

Nos itérations sont plus longues que celles prévues par XP

Client sur site

Pour le noyau, le client c'est nous
(inclus communauté co-développement)

Pas encore expérimenté avec client extérieur

Causes:

- Distance (centres de recherche a l'étranger)
- Connaissance imparfaite de la méthode

Spatiopole, rue des Chasseurs Ardennais, B-4031 Angleur, Belgium Tél. (+32)4/37.29.345 www.open-engineering.com

Rythme durable

Pas toujours évident dans une jeune PME !!!

En moyenne 45h/semaine

Flexibilité des horaires favorisée
(production de code = activité créative)

Spatiopole, rue des Chasseurs Ardennais, B-4031 Angleur, Belgium Tél. (+32)4/37.29.345 www.open-engineering.com

Résumé

- Conception simple & refactoring…… ✓
- Tests unitaires et de recettes…… ✓
- Responsabilité collective du code…… ~
- Programmation en binôme…… ✗
- Règles de codage…… ✓
- Métaphore…… ✓
- Intégration continue…… ✓
- Livraisons fréquentes…… ~
- Planification itérative…… ✓
- Client sur site…… ~
- Rythme durable…… ~

Par où commencer ?

Programmation par tests (unitaires et recettes)
Intégration continue
Refactoring
Programmation itérative

Références :

Livre : « L'extreme programming » (Eyrolles)
Web : http://www.xprogramming.com
http://industrialxp.org