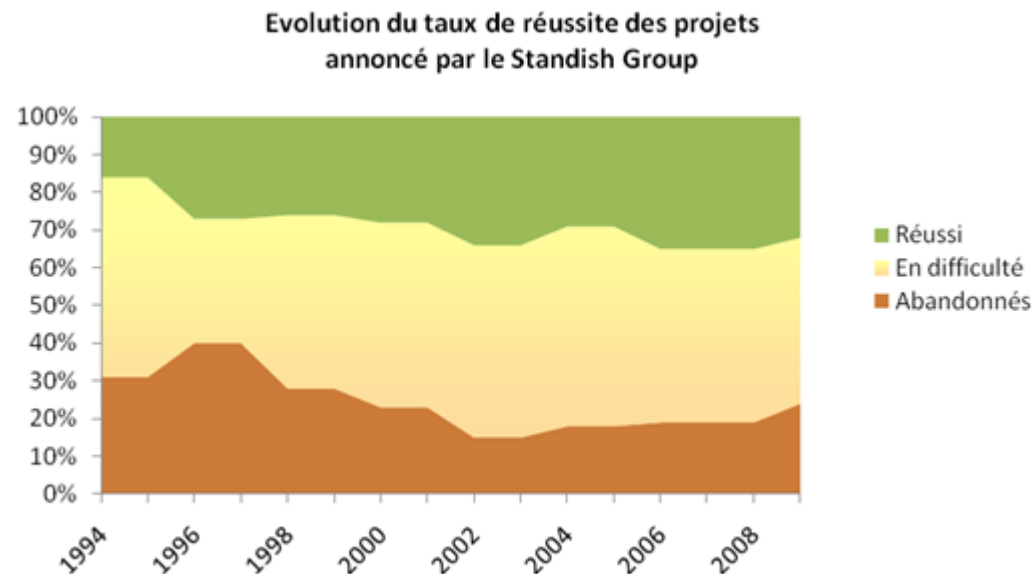


Méthodes agiles

ICAM – JP Gouigoux – 11/2012

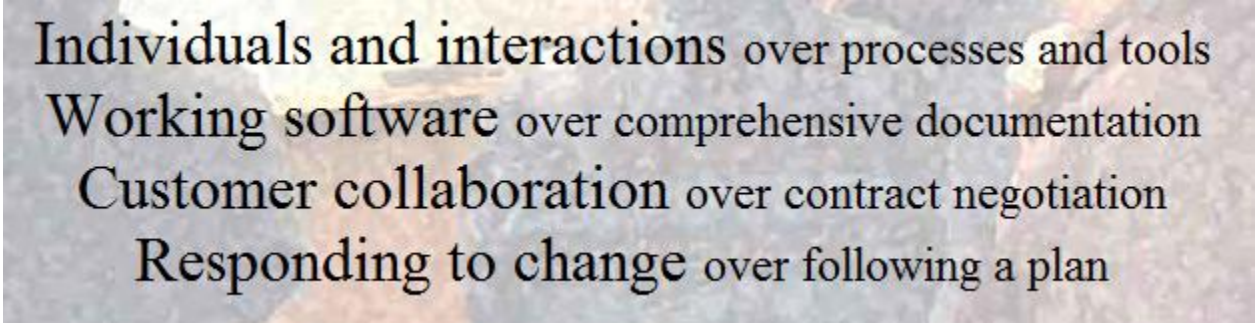
Un peu d'histoire

- Développement industriel très jeune
 - Merise / cycle en V : premières méthodes
 - Majorité des projets en échec



Le manifeste agile

- Méthodes venant de la base



Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

Un vrai mouvement



IMAGINE AN AGILE WORLD

Conferences

- Call For Speakers
- Result of Speakers Submissions

-  Sydney
-  Vienna
-  Brussels
-  Namur
-  Aracaju
-  Belo Horizonte
-  Brasília
-  Belém
-  Campo Grande
-  Curitiba
-  João Pessoa
-  Juazeiro do Norte
-  Maringá

Welcome to Agile Tour 2012!

Agile Tour, a series of non-profit events over several cities throughout October and December 2012.



Adoption progressive en cours

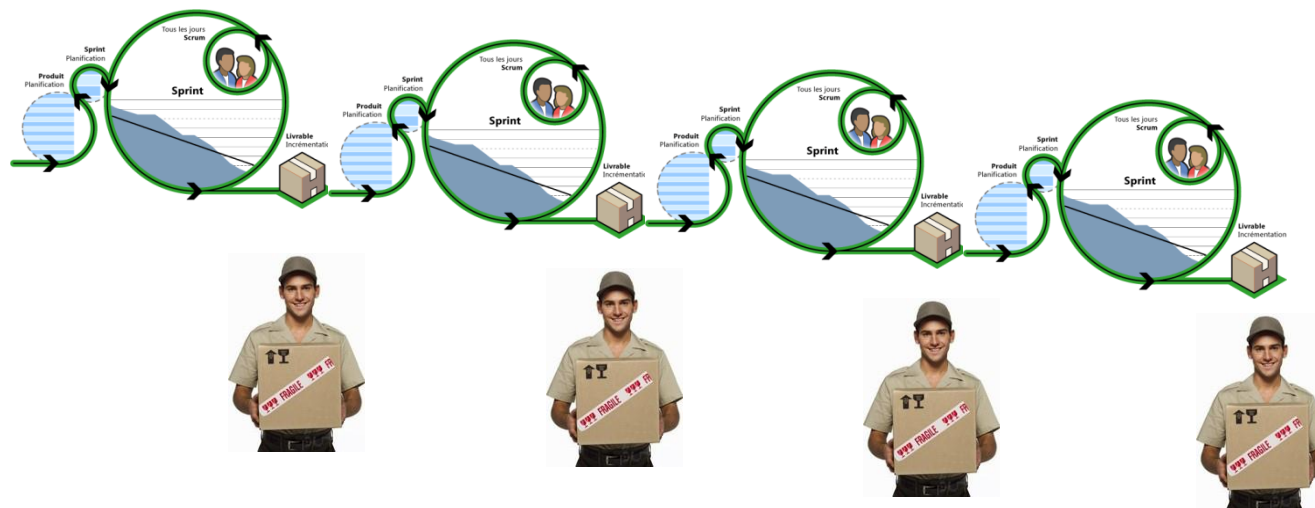
- Plus de la moitié des sociétés d'éditions informatiques ont des projets agiles
- ROI démontré
- Certaines parties de la méthode peuvent être utilisées seules
 - TDD, Pair Programming
 - Daily stand-up meetings, artefacts SCRUM
- Mais le vrai gain vient de l'adoption de la philosophie complète

Au temps des cavernes



4 mois

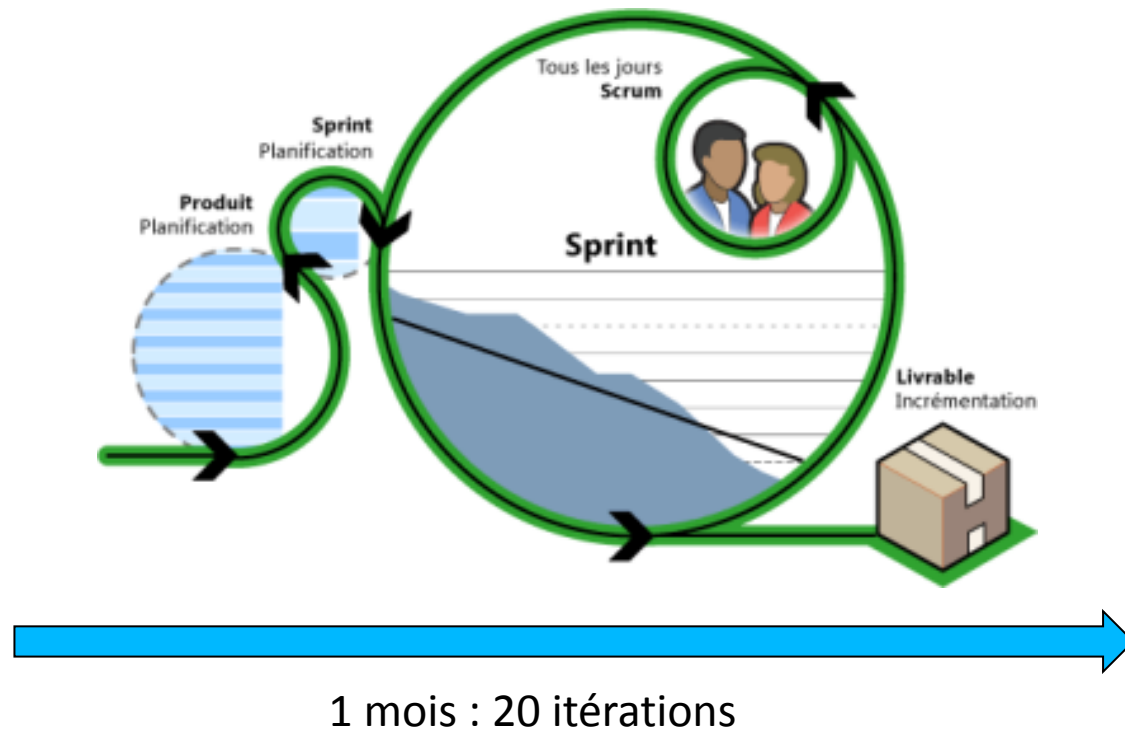
Après dix ans de travail



4 mois



Un « sprint » au sens SCRUM



Importance des tests

- Definition of Done
 - Analysé
 - Codé
 - Testé
 - Documenté
 - Livré
 - Validé
- Seule l'automatisation complète du cycle permet de réaliser des itérations rapides

Entrées : Nom du pack, Version, Mode Debug, Inclure les tests longs, [Composants à traiter]

Entrées : RapportTests.xml, Composants à traiter, Inclure

Entrées : InIsb.config, RepComposantsDebug ou RepComposantsRelease

Entrées : Répertoire pour les DLL, répertoire pour le rapport, RapportTests.xml, Restrindre la liste des Composants, Inclure les tests longs.

Entrées : InIsb.config, CheminFichierConfiguration

Entrées : InIsb.config, FichierConfigurationNUnit

Aucune config n'a été trouvée dans le fichier.

Entrées : Répertoire pour les DLL, répertoire pour le rapport, Restrindre la liste des Composants, Chemin d'accès au fichier xml correspondant au pack, le nom de la config à tester, le mode de test.

Sorties : Vrai si le test NUnit est correct, Faux s'il y a une erreur dans ce test.

Sorties : Vrai si toutes les valeurs renvoyées par LancerTestNUnitParConfig sont vraies

Sortie : Le log XML

Paramètres : /xml : Résultats.xml /output : stdout.txt /err : stderr.txt /exclude : Voir test long

Début : ExecuterNUnitPack [NOS_Build]

Blocage du composant par l'application.

Début : LancerTestsNUnitRepTemp [Pack]

Création des répertoires pour les DLL et pour le rapport

Récupération de la composition du pack

Remplissage initial du pack

Début : LancerTestsNUnit [Pack]

Initialisation

Recherche du fichier xml de configuration correspondant au pack.

Recherche du fichier de configuration NUnit

Pour chaque configuration présente dans le fichier de config NUnit

Début : LancerTestsNUnitParConfig [Pack]

Traitement

Fin : LancerTestsNUnitParConfig [Pack]

Fin de la boucle sur les configurations

Intégration des résultats des tests Unitaires dans un seul fichier

Envoi d'un email détaillant les erreurs s'il y en a.

Fin : LancerTestsNUnit [Pack]

Fin : LancerTestsNUnitRepTemp [Pack]

Déblocage du composant par l'application.

Fin : ExecuterNUnitPack [NOS_Build]

Blocage du composant par l'application

Composant Disponible

Non

Oui

Composant en cours d'utilisation.

On prends la main sur le composant

Remplissage initial du pack

Boucle sur les composants du pack

Recherche du répertoire du composant

Recopie des répertoires du composant.

Fin de la boucle

Récupération des Tests Unitaires des composants

Pour chacun des trois modules : AspNet_11, AspNet_20, ClickOnce

Création d'un répertoire pour les tests

Module de type AspNet

Oui

Non

Recopie du répertoire **Modeles** pour le fonctionnement de certains

On autorise la Lecture/Ecriture des fichiers xml présents dans **IDonneesTests**

Fin de la boucle

Lancement des tests Unitaires sur les configurations

Récupération de la liste des composants du pack

Pour chacun des trois modules : AspNet_11, AspNet_20, ClickOnce

Création d'un fichier de test unitaire NUnit

Pour chaque composant du pack

Incorporation des tests Unitaires du composant dans ceux du pack.

Fin boucle sur les composants

Il y a au moins un test

Oui

Non

Inclure les tests longs

Oui

Non

Paramètre /exclude : Long

On lance NUnit-console.exe

Fin de la boucle sur les modules

Récupération des Tests Unitaires des composants

Composant à traiter

Non

Oui

Fin

Recherche de tous les projets de type test présents dans le fichier XML de description du composant

Pour chacun de ces projets tests

Projet actif, livrable, de même module que celui en cours

Non

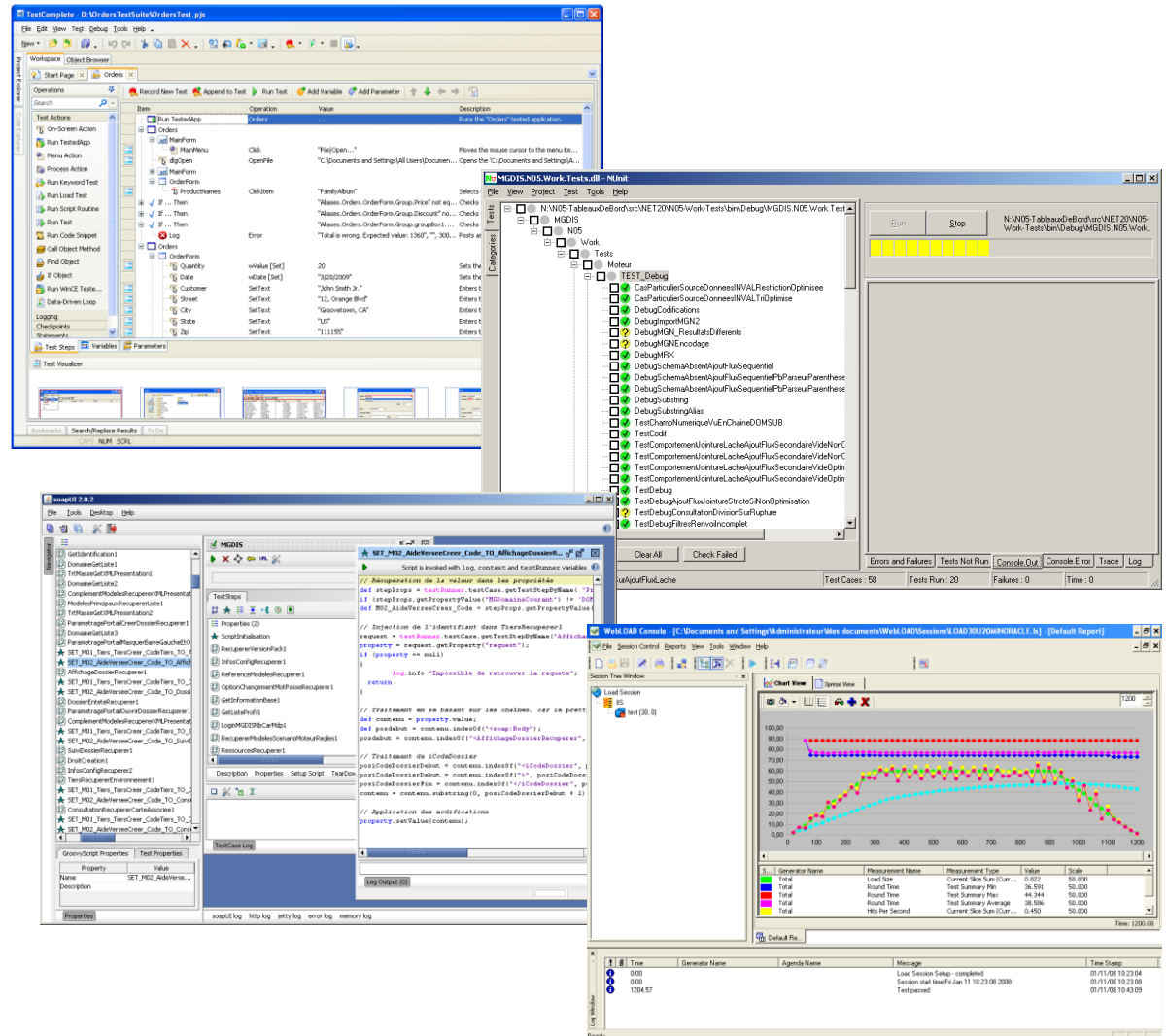
Oui

Fin

On ajoute l'assembly du test du composant aux tests du pack

Fin boucle sur les projets

- Unitaires
- Intégration
- Régression
- Rejeu client
- Robustesse
- Performance
- Charge



Même des outils spécialisés

- Sur mesure
- Application à auto-diagnostic

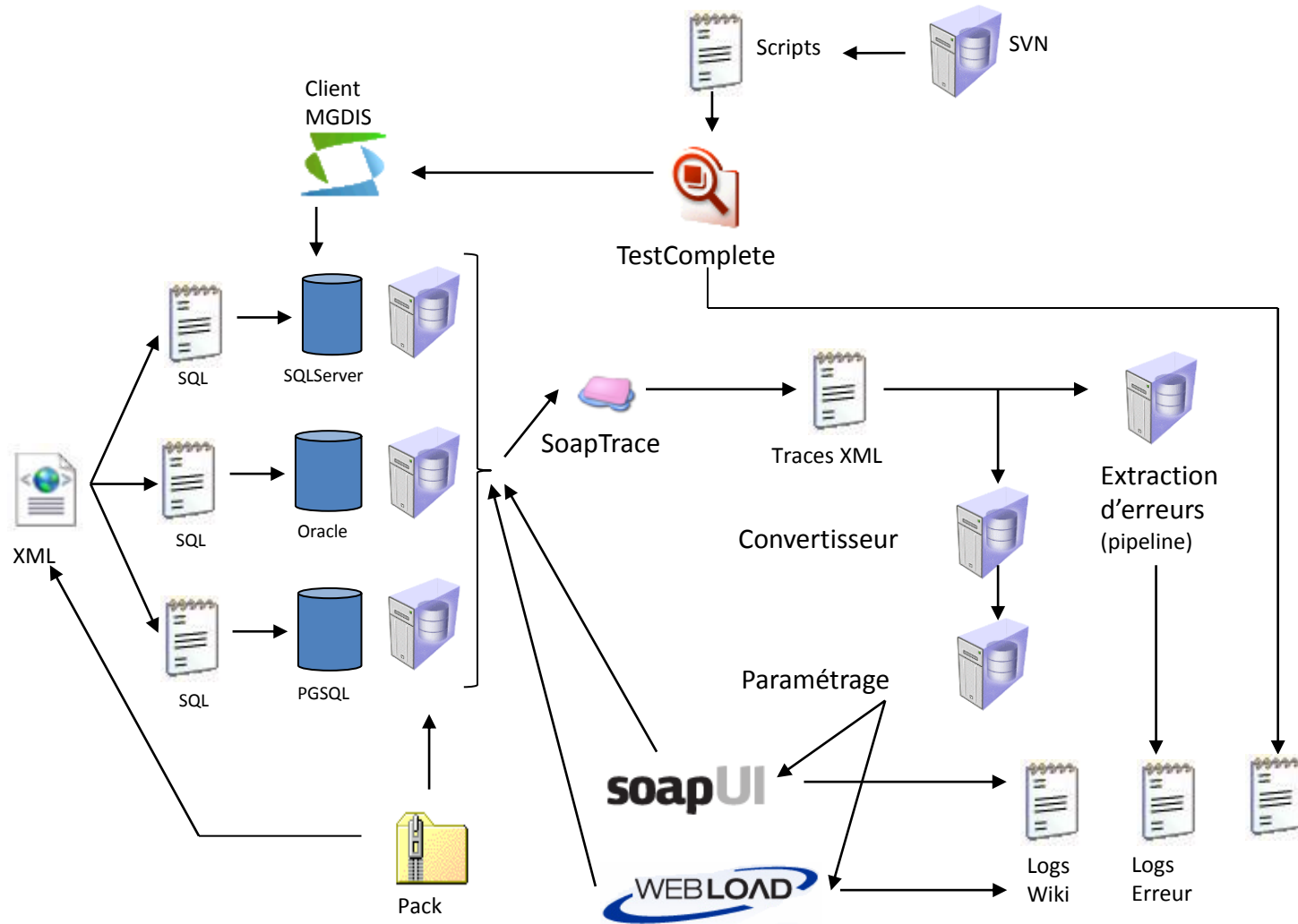
	2001	2002	2003	2004	2005
Montant	100000	0	0	0	0
Durée	4	0	0	0	0
Mode	3	0	0	0	0
Début	01/01/2001	01/01/2002	01/01/2003	01/01/2004	01/01/2005
Taux	0	0	0	0	0
Capital amorti	31250	25000	25000	18750	0
Restant à amortir	68750	43750	18750	0	0

0 : linéaire
1 : SOYD croissant
2 : SOYD décroissant
3 : dégressif
4 : croissant avec taux

OK

Sauvegarder ce calcul comme référence de test unitaire (le résultat doit être valide)

Tout en automatique la nuit



- Intranet local 100%

Outils complémentaires

The image displays three overlapping screenshots of Visual Studio, illustrating various development tools and their integration.

Top Left: Code Coverage
The screenshot shows the 'Coverage Explorer' window, displaying a tree view of the project's code coverage. The 'Program.cs' file is selected, showing a list of methods and their coverage status. The 'Program.cs' file is highlighted, showing a list of methods and their coverage status.

Top Right: Codeplex-DLR - Microsoft Visual Studio
The screenshot shows the 'Codeplex-DLR' window, displaying a dependency graph for the 'PdnLib' assembly. The graph shows the relationships between various assemblies, including 'PaintDotNet.Effects', 'PaintDotNet.Data', 'PdnLib', 'PaintDotNet.Resources', 'PaintDotNet.SystemLayer', 'PaintDotNet.StyleReader', 'System.Drawing', 'System.Windows.Forms', 'System.Xml', and 'System'. The 'PdnLib' assembly is highlighted, showing its dependencies on other assemblies.

Bottom: Microsoft FxCop - Softpedia*
The screenshot shows the 'Microsoft FxCop' window, displaying a list of rules and their status. The 'Usage Rules' rule is selected, showing its details. The 'Usage Rules' rule is highlighted, showing its details.

Dependency Graph Data
The dependency graph shows the following assemblies and their dependencies:

- PaintDotNet.Effects (0, 1, 2, 3, 4, 5)
- PaintDotNet.Data (1, 5, 136, 66, 16)
- PdnLib (2, 156, 100, 23, 102)
- PaintDotNet.Resources (3, 7, 5, 12, 2)
- PaintDotNet.SystemLayer (4, 19, 63, 3, 2)
- PaintDotNet.StyleReader (5, 2, 2, 2, 2)
- mscorlib (6, 93, 161, 296, 73, 204, 25)
- System.Windows.Forms (7, 158, 166, 220, 7, 96, 2)
- System.Drawing (8, 68, 116, 396, 8, 60, 9)
- System (9, 11, 16, 52, 3, 26, 2)
- System.Xml (10, 7, 7, 7, 7, 7)

FxCop Rule Details
The 'Usage Rules' rule is selected, showing its details:

- Checked: True (Boolean)
- Children: Count == 1 (NodeBaseDictionaryCollection)
- Container: Count == 9 (NodeBaseDictionary)
- DefaultCheckState: True (Boolean)
- DisplayName: Usage Rules (String)
- FileIdentifier: c:\program files\microsoft fxcop 1.36\Rules\UsageRules.dll
- FileName: c:\program files\microsoft fxcop 1.36\Rules\UsageRules.dll

Extreme Programming - Valeurs

- Simplicité
- Communication
- Courage
- Feedback
- Respect

Extreme Programming - Pratiques

- Intégration continue
- Rythme soutenable à long terme
- Métaphores
- Refactorisation
- Propriété commune du code
- Pair Programming
- Client final accessible

SCRUM - Pratiques

- Méthode de conduite de projet
- Meetings
 - Daily stand-up
 - Backlog
 - Planification de sprint
 - Revue de sprint
 - Retrospective

SCRUM – Principes

- Rôles
 - Product Owner
 - Scrum Master
 - Développeur
- Règles
 - Sprint fixé / délai fixé
 - Scope variable
 - Vitesse mesurée

Quelques chiffres

- Livraison bloquée sauf si :
 - 100 % des 8000 tests unitaires OK
 - 100 % des 120 scénarios TestComplete OK
- Couverture du code par les tests
 - 80 % sur les API
 - 20 % sur le métier
- Performance améliorée de 2% par version

Le secret de l'agilité

- Ne pas faire confiance à la méthode
- Créer sa propre méthode
- Itérer

