

Master SIS
Professionnel

Option Génie Informatique

Atelier de Génie Logiciel

Illustration avec Windev

[UE GL 7)

Spécialité Génie Logiciel

Erwan TRANVOUEZ

erwan.tranvouez@polytech.univ-mrs.fr

Université
Paul CEZANNE

Planning de sessions

- 3 séances de cours
- 2 séances de TP
- 1 séance de cours
- 2 séances de TP
- 1 cours
- 1 séance de TP

- 1 séance en réserve Cours/TP planifiée

- Notation :
 - Mini Projet de Conception/développement Web avec Windev
 - Conception en 2 temps : prévisionnel simplifié / recette
 - Soutenance/démonstration
 - => note individuelle

Ressources (Atelier de) Génie Logiciel

- *Software Engineering : Principles and Practice*. **H. V. VLIET**. 2nd ed. J. Wiley & sons Ltd.
- *Analyse de système orientée-objet et génie logiciel: Concepts, méthodes et application*. **G. LEVESQUE**. Chenelière/Mc Graw-Hill.
- *Génie Logiciel*, Jacques PRINTZ Que Sais-Je N° 2956. PUF.
- *Ingénierie des Systèmes d'Information : MERISE*. **D. NANJI, B. ESPINASSE et al.** 4^{ème} ed. Vuibert.
- *Guide to CASE adoption* **K.S. OAKES, D. SMITH, E. MORRIS**, Tech. Report Software Engineering Institute - Carnegie Mellon Univ.
- *Et de manière plus générale :*
http://www.sei.cmu.edu/legacy/case/case_what_is.html
- *Xtreme Programming :*
 - <http://www.extremeprogramming.org/>
 - <http://c2.com/cgi/wiki?ExtremeProgramming>

Ressources Windev & Outils

- *Liste d'AGL par catégorie :*
<http://www.cs.queensu.ca/Software-Engineering/toolcat.html>
- *Documentation commerciale des produits mentionnés.*
 - <http://www.pcsoft.com/windev>
 - <http://www.sybase.com/products/internetappdevtools/powerbuilder>
 - <http://www.rational.com/products/dstudio>
 - <http://www.oraclecom/ip/develop/ids/editions.html>
 - <http://www.objecteering.com>
- *Guide d'AutoFormation.* PC Soft Edition.
- *Site de l'Association des développeurs Windev.*
<http://www.windevasso.org>

Plan du support de cours

- I** (Atelier) Génie Logiciel : Introduction
 - II** Premiers Pas avec Windev
 - III** Programmation procédurale avec Windev : le W-Language
- IV** Programmation graphique avec Windev
- V** Développement d'Applications gérant des Bases de Données
 - VI** Compléments

I - (Atelier de) Génie Logiciel : Introduction

- 1. (Rappels) sur le Génie Logiciel**
- 2. Qu'est ce qu'un Atelier de Génie Logiciel**
- 3. Présentation générale de Windev**

Contexte

Génie Logiciel propose :

- **Méthodes** (gestion de projet, conception, développement, ...)
- **Formalismes** (UML, SADT, Z, ...)
- **Techniques** (normes, ...)

Atelier de Génie Logiciel propose :

- *Assistance au*
- *automatisation du*
- *Systématisation du*

**GL dans tout le cycle de vie
d'un projet informatique**

(Rappels) sur le Génie Logiciel

Définition du terme **Logiciel (Software)**:

*Le logiciel est l'ensemble des programmes, procédés et règles, et éventuellement de la **documentation**, relatifs au **fonctionnement** d'un ensemble de **traitement de l'information**.*

Définition du terme **Génie Logiciel (Software Engineering)**:

- *Le Génie Logiciel (GL) est l'ensemble des activités de **conception** et de **mise en œuvre** des produits et procédures tendant à **rationaliser** la **production** du logiciel et son **suivi**.*
- *L'art de **spécifier**, de **concevoir**, de **réaliser**, et de **faire évoluer**, avec des **moyens** et dans des **délais raisonnables**, des programmes, des documentations et des procédures de qualité en vue d'utiliser un ordinateur pour résoudre certains problèmes.*

Le cycle de vie d'un logiciel : modèle en cascade

Problème

« WaterFall Model » attribué à Royce. Il définit **clairement** les différentes étapes du cycle de développement d'un logiciel...

Analyse des besoins
[Faisabilité]
V & V

Spécification
V & V

Implémentation
V & V

Test
V & V

Maintenance
V & V

... mais il demeure idéaliste voire irréaliste :

- le coût des allers-retours est sous-estimé.
- difficulté pour effectuer l'analyse des besoins : problème de communication Client ↔ Informaticien

V & V :

Vérification : le système répond aux besoins identifiés à la phase précédente.

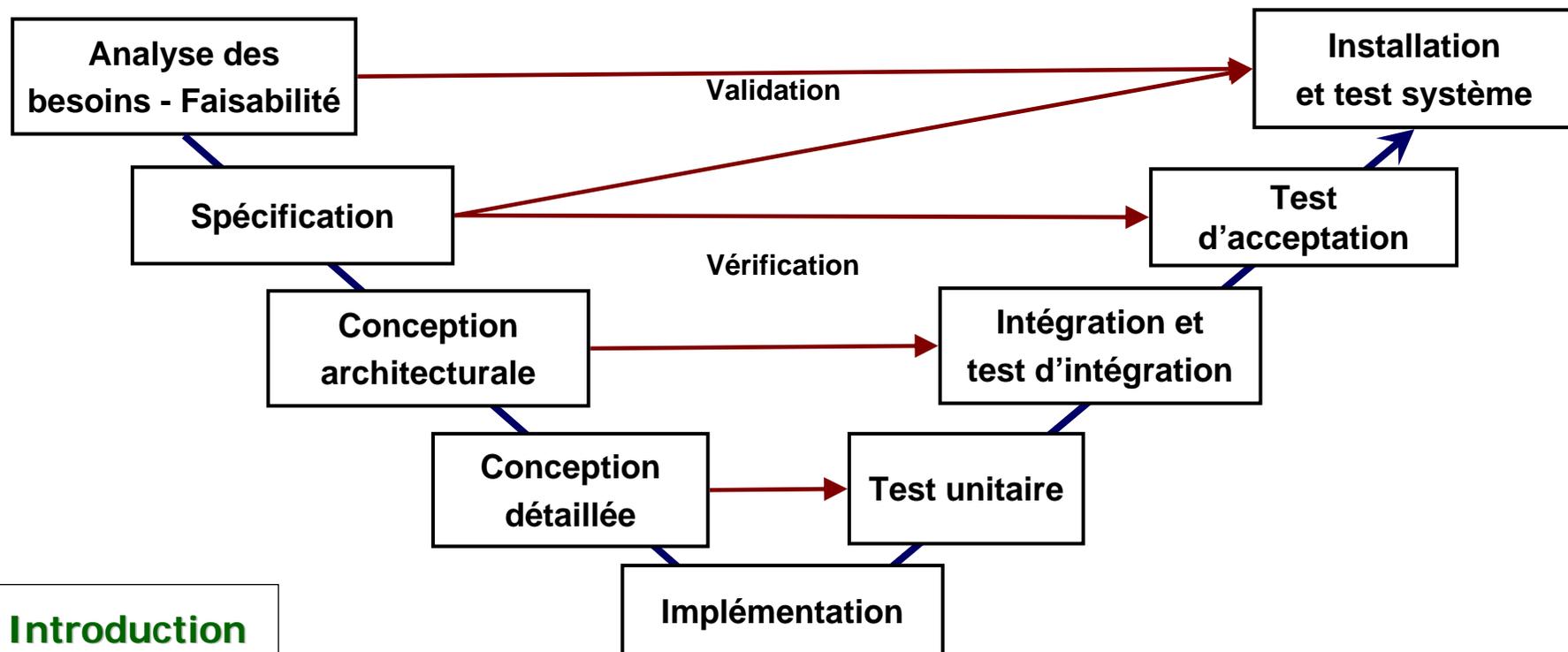
Validation : le système réponds aux besoins du client.

Tests unitaires
Tests d'intégration

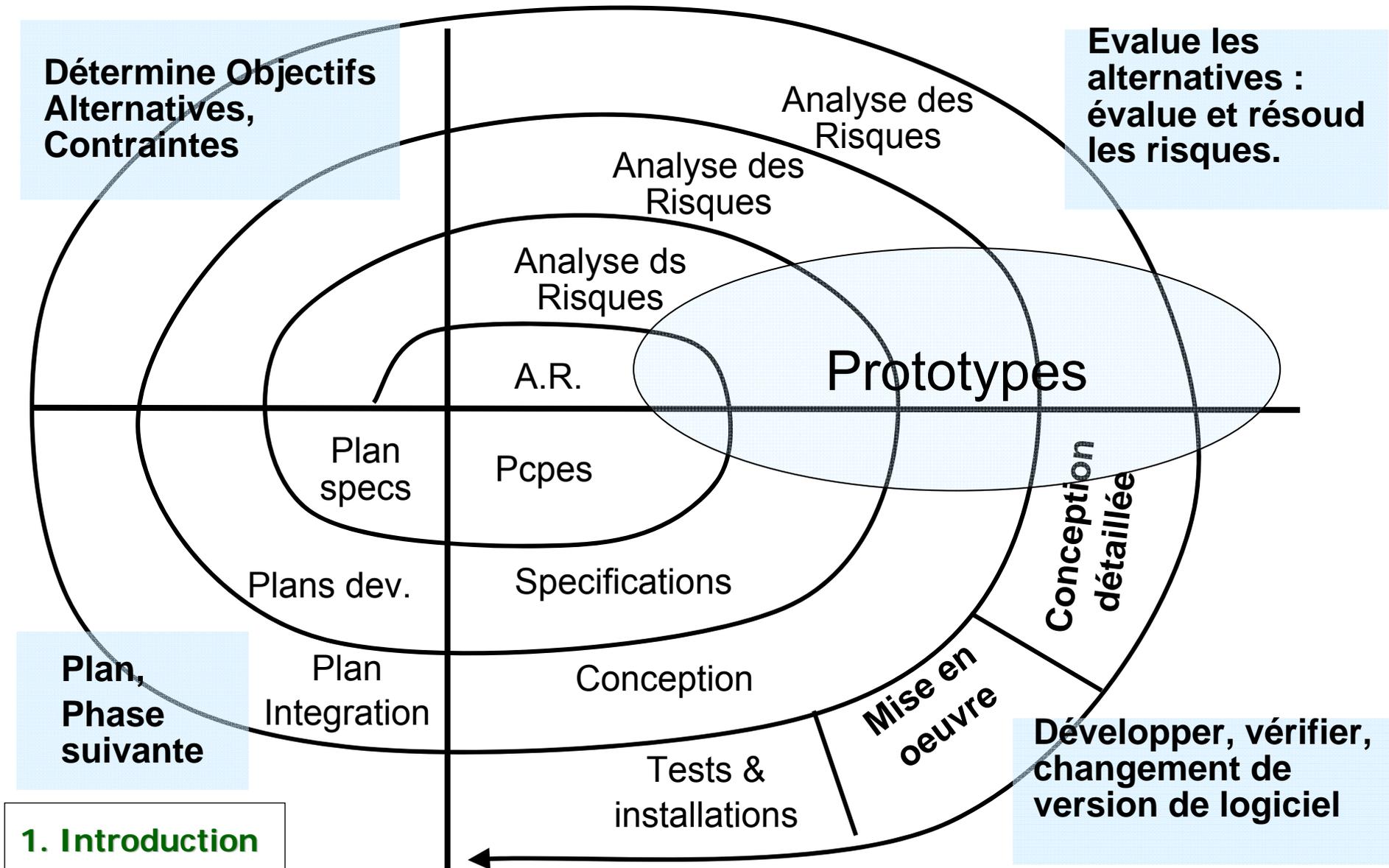
Mise à jour, correctifs
Gestion des configurations

Le cycle de vie d'un logiciel : modèle en V

- Les premières étapes préparent étapes finales (**ex : lors de l'analyse des besoins on propose des tests de validation**)
- Toute activité de décomposition doit prévoir la phase de recombinaison (**amélioration de l'intégration**)
- Cependant : hypothèse de séquentialité forte, prise en compte du maquettage absente, pas de prise en compte de l'évolution du cahier des charges.



Le cycle de vie d'un logiciel : modèle en Spirale



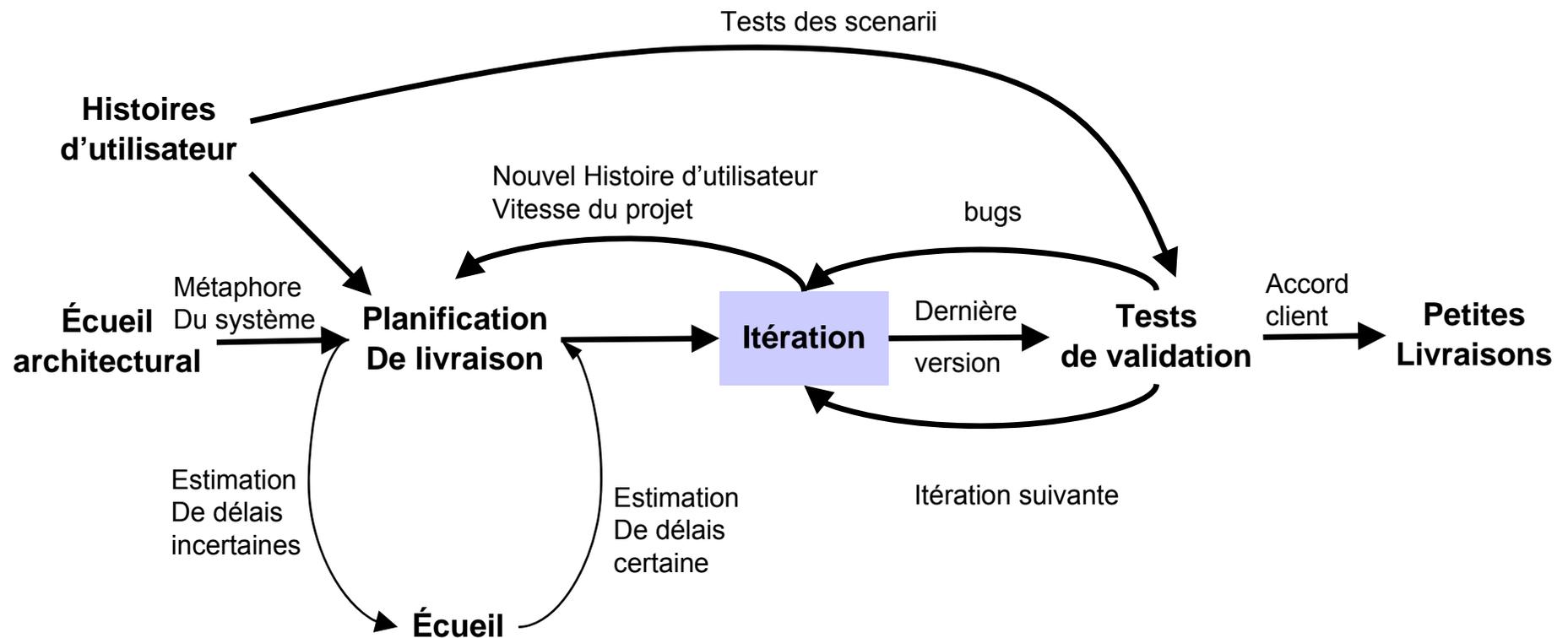
Xtreme Programming : présentation générale

- Approche alternative au développement d'applications logicielles.
- Prend le contrepied des autres méthodologies (qualifiées de *BigDesignUpFront*) en réduisant l'importance de la phase d'analyse et de conception au minimum.
- Suit la logique des méthodologie de type RAD (Rapid Application Development)
 - ⇒ *Efficace sur des projets de taille petite à moyenne...*
 - ⇒ *Gestion des risques (évolution de l'expression des besoins)*
- L'objectif final est le **code** : un programme qui répond aux besoins du clients, testé et structurellement le plus simple possible. La complexité est gérée par des cycles itératifs.
- Intègre largement la dimension humaine du projet.
- Résultat : propose une série de principes, de pratiques : presque une philosophie plutôt qu'une méthode.

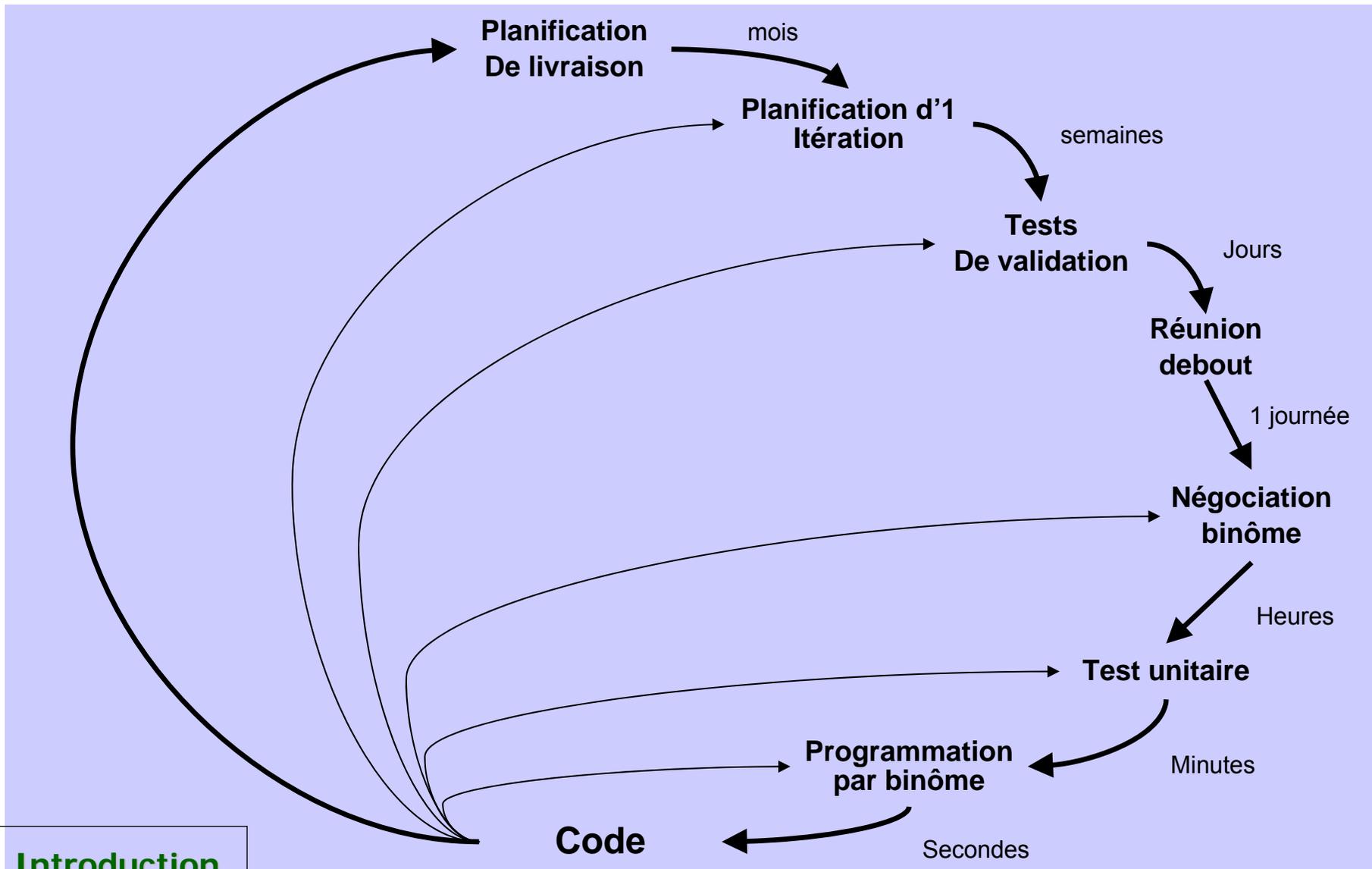
Xtreme Programming : Principes

- **Simplicité** : aller au plus simple et complexifier si nécessaire (donc ne pas aller tout de suite au maximum de généralité du code ou de l'architecture. Il le sera si nécessaire au prochain cycle de production)
- **Communication** : entre tous les acteurs. Concernant le client l'usage de *métaphores* (barrière de langage client ↔ développeur), d'*histoires* (user stories) est recommandé.
- **Retours d'informations** (feedback) de tous les acteurs du processus
- **Développement tiré par les tests** : écrit dès le début.
- **Développement tiré par les besoins du clients** : *user story* exprimé à l'aide de cartons.
- **Refactoring** ou reconception du code... Une fois qu'il marche réécriture du code pour le simplifier.
- Etc.

Xtreme Programming : Gestion de projet



Xtreme Programming : Itérations ...



Cas particulier SI : le CDM d'Oracle 1/2

- **Exemple de méthode de réalisation de SI proposée par les services consulting d'éditeurs de logiciels ... intégrant leurs produits.**

- **Oracle CDM - Custom Development Method**

- partie *développement* de la méthode globale d'ingénierie des SI *Oracle Method*SM (aspects stratégiques, business reengineering, Gestion de projet etc.)

- couvre le cycle de vie de développement de SI Informatisé.

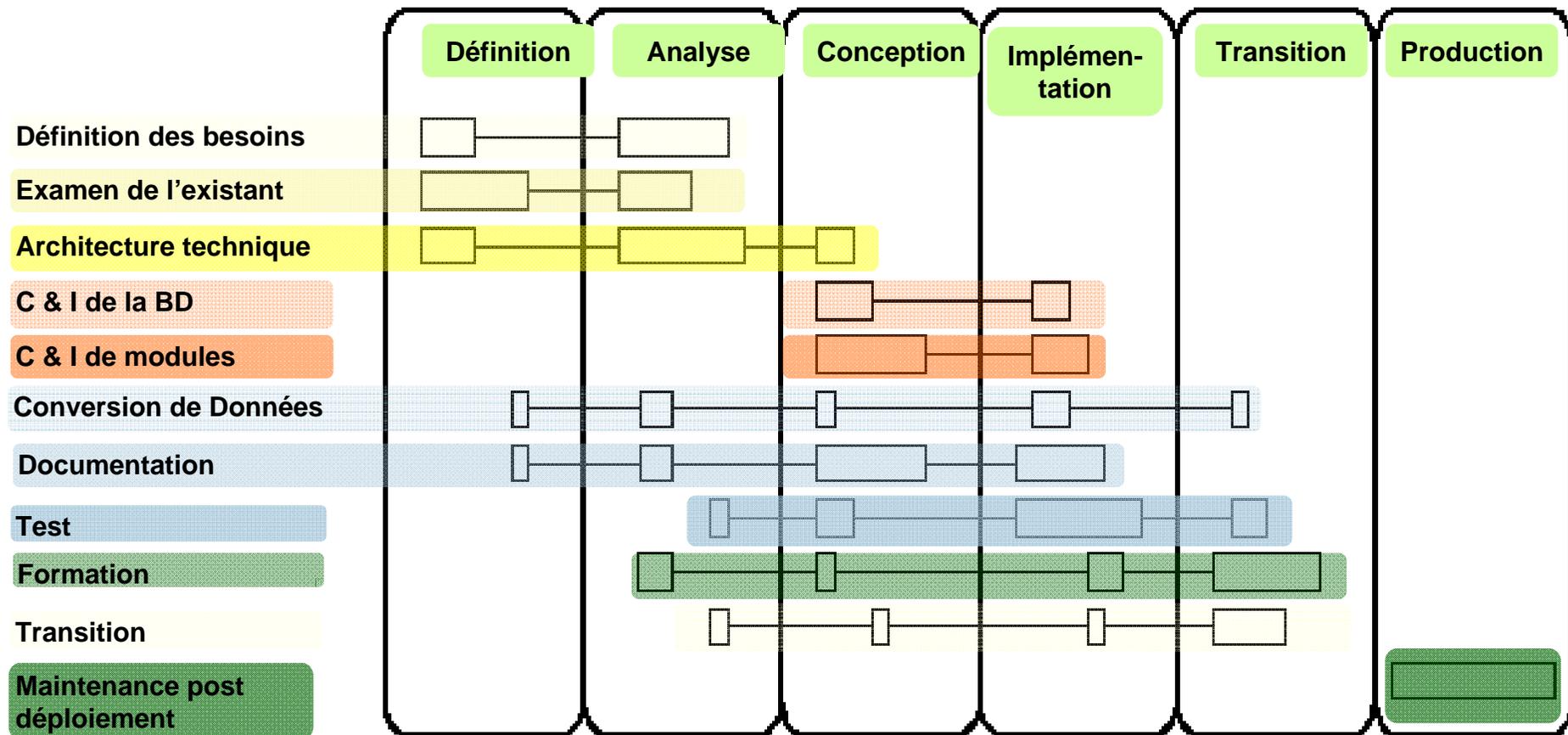
- basé sur une démarche systémique

- **Décliné sous deux dénominations :**

- CDM Classic : approche linéaire en phases distinctes

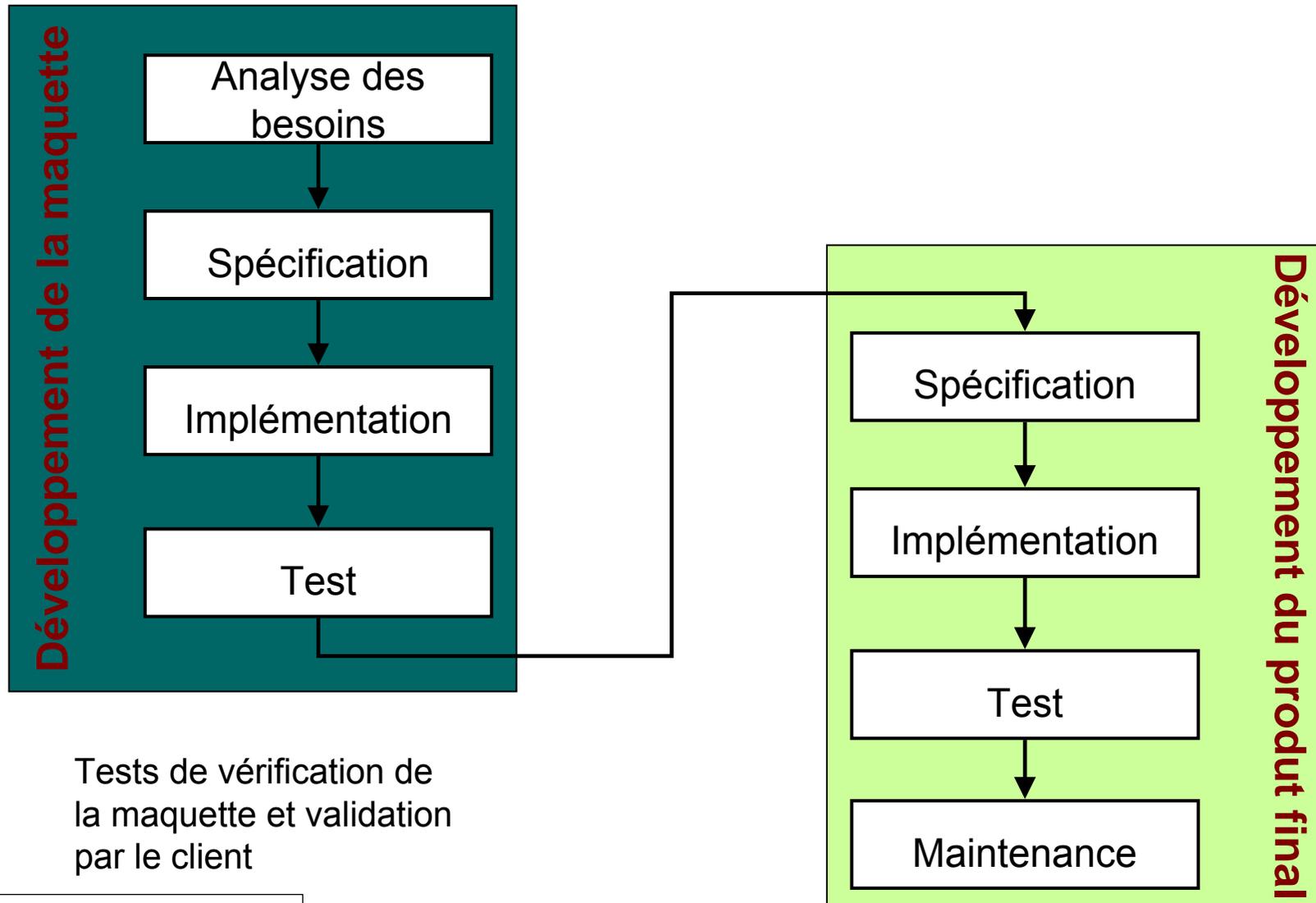
- CDM Fast Track : un concept RAD (Rapid Application Development) basé sur DSDM (Dynamic Systems Development Method) postulant le prototypage, le développement et la fourniture d'artefacts itératifs et incrémentaux, le timeboxing (jalonnement), l'analyse de la valeur et une validation continue.

Cas particulier SI : le CDM d'Oracle 2/2



C & I = Conception et Implémentation

La solution du maquettage 1/2



Tests de vérification de la maquette et validation par le client

La solution du maquettage

L'analyse des besoins est effectuée via la maquette :

- Requier un langage/des outils de développement de haut niveau pour pouvoir développer très rapidement...
- ... une version simplifiée, avec moins de fonctionnalité,
- soulève immédiatement les erreurs d'interprétation des besoins du client.

2 types de stratégies :

- Prototype jetable : une fois validé, on ne garde du prototype que la coquille : tout est redéveloppé (ex: autre langage et autres technologies).
- Prototype incrémental : le développement se poursuit sur les mêmes bases.

Qu'est ce qu'un Atelier de Génie Logiciel ?

Définition du terme **Atelier de Génie Logiciel (AGL)** :

Outil informatique aidant à la production d'un logiciel. Il assiste la démarche de Génie Logiciel poursuivie.

Le terme anglais est plus explicite **CASE tools (Computer Aided Software Engineering)**.

Objectif : Automatisation maximale du processus d'une partie ou tout le processus de développement du logiciel ...

... objectif pondéré par la réalité du terrain.

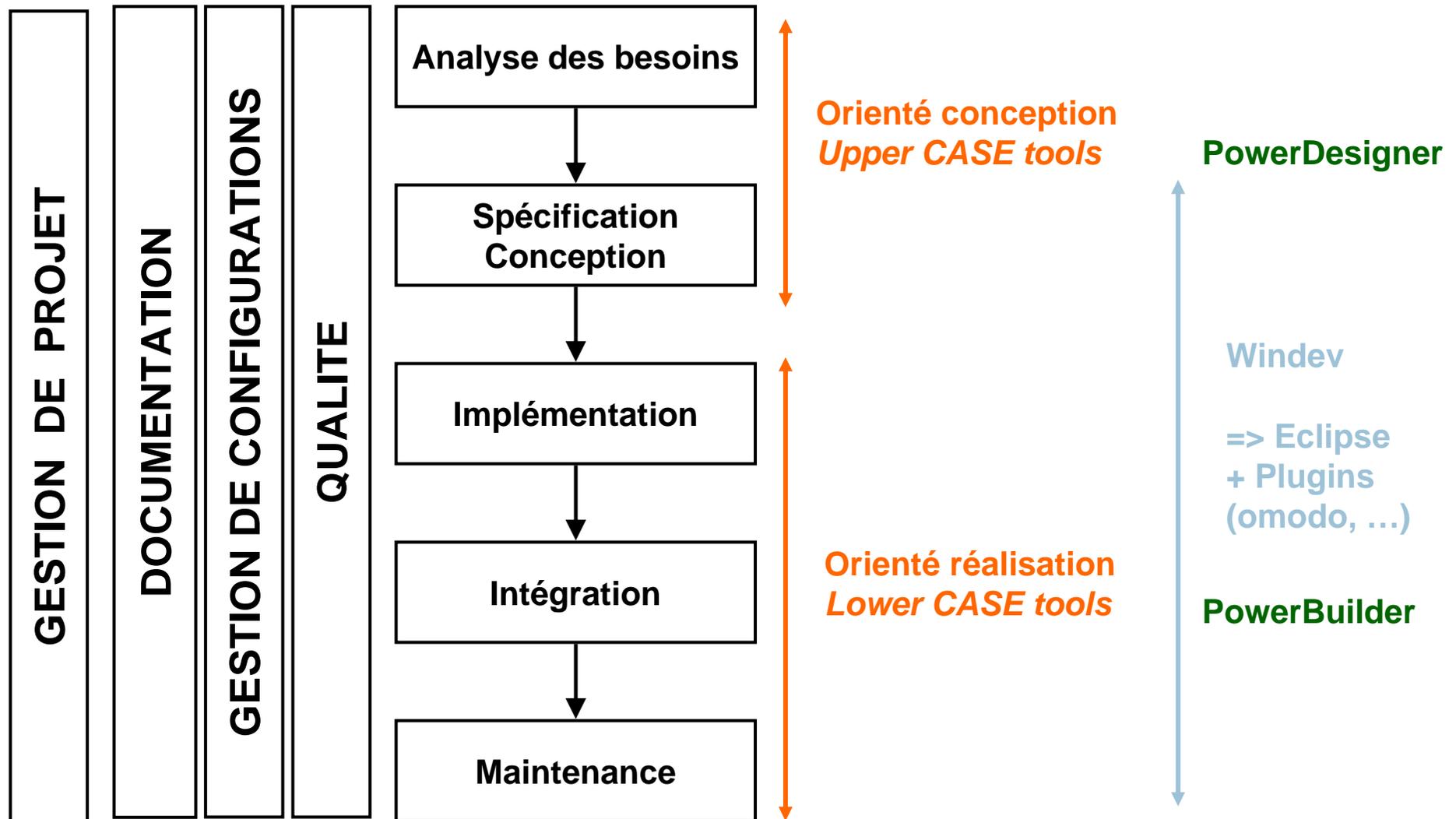
=> Implique d'assister les phases du cycle de vie du logiciel

Catégories d'AGL

Les AGL peuvent être classés selon plusieurs aspects :

- **richesse du support** : ensemble d'outils, outils intégrés, aide à la démarche.
- **type de problèmes** : logiciels embarqués, temps réel, "business applications", applications métiers ...
- **type de projet d'ingénierie logicielle** : développement logiciel (cf. cycle de vie), intégration de systèmes, système à base de connaissance.
- **Ampleur du projet** : complexité, nombres de participants, durée ...
- **Gestion des ressources du projet** : les considérations managériales des ressources mises en œuvre dans le projet sont elles prises en compte ? (planification, ordonnancement, ...).
- **Phase du cycle de développement prises en compte** : conception et/ou développement.

Classification basée sur le cycle de développement



Fonctions transversales d'un AGL

- Participent à toutes les phases de développement et maintiennent la cohérence entre les phases.

Activité	Type d'outils
Gestion de processus GL	Sorte Meta-AGL : en fonction du type de méthodologie retenue (RUP, Merise,...) l'outil assiste la mise en œuvre de la méthode (<i>IBM Rational Method Composer</i>) // avec SPEM.
Gestion de projet	Outils de Planification, calculs de coûts, suivi de projet (ex MS Project), de tableurs, ...
Outils collaboratifs	Gestion utilisateurs, droits accès, communication, ...
Documentation	Editeurs de texte (code, aide, ...) d'images, diagramme etc...
Gestion de Configurations	Etendu aux autres productions (documentation, diagramme, ...)

AGL Orienté Conception

- **Assistent la phase initial du projet de développement.**
- **Fortement basé sur des paradigmes (Orienté Objet), des méthodes de conception et les formalismes associés (ex : RUP/UML, Merise/E-R, ...).**
- **Proposent des outils d'éditeurs graphiques de ces formalismes.**
- **Proposent une assistance pour la génération de documentation.**
- **Peuvent proposer un outil de prototypage (génération automatique partielle de code)**
- **et éventuellement de *reverse engineering* (création de représentations graphique dans un formalisme donné à partir de code source existant).**

Fonctions assistant la conception

- Participent à toutes les phases de développement et maintiennent la cohérence entre les phases.

Activité	Exemples
RéIngénierie	Analyse de l'existant, reverse Engineering,
Edition	Surtout graphique pour exprimer des concepts (expression des besoins) ou mettre en oeuvre les formalismes des méthodes sélectionnées.
Prototypage	Générateurs d'IHM, génération de code automatisé ou assisté, langage haut niveau, interpréteurs ou compilateurs dédiés
Support méthodologique	Editeur graphique intelligent (vérification/validation diagramme par ex), dictionnaire de donnée

Quelques AGL Orientés Conception

- **Win'Design de CECIMA.**
- **PowerDesigner de Sybase.**
 - Basés sur Merise et UML (spécialisé dans le développement de SI).
- **Oracle Designer d'Oracle Corporation.**
- **Rational Suite AnalystStudio, Rational Rose.** Basés sur UML (toute application basée sur l'approche objet voir également développement SI) + **Rational Rose Unified Process** (support méthodologique).
- **Objecteering de SoftTeam.** Basé sur UML.
- **Eclipse** ~~bientôt ?~~ (<http://www.eclipse.org>) (ex. plugin UML, ...)

AGL Orienté Réalisation

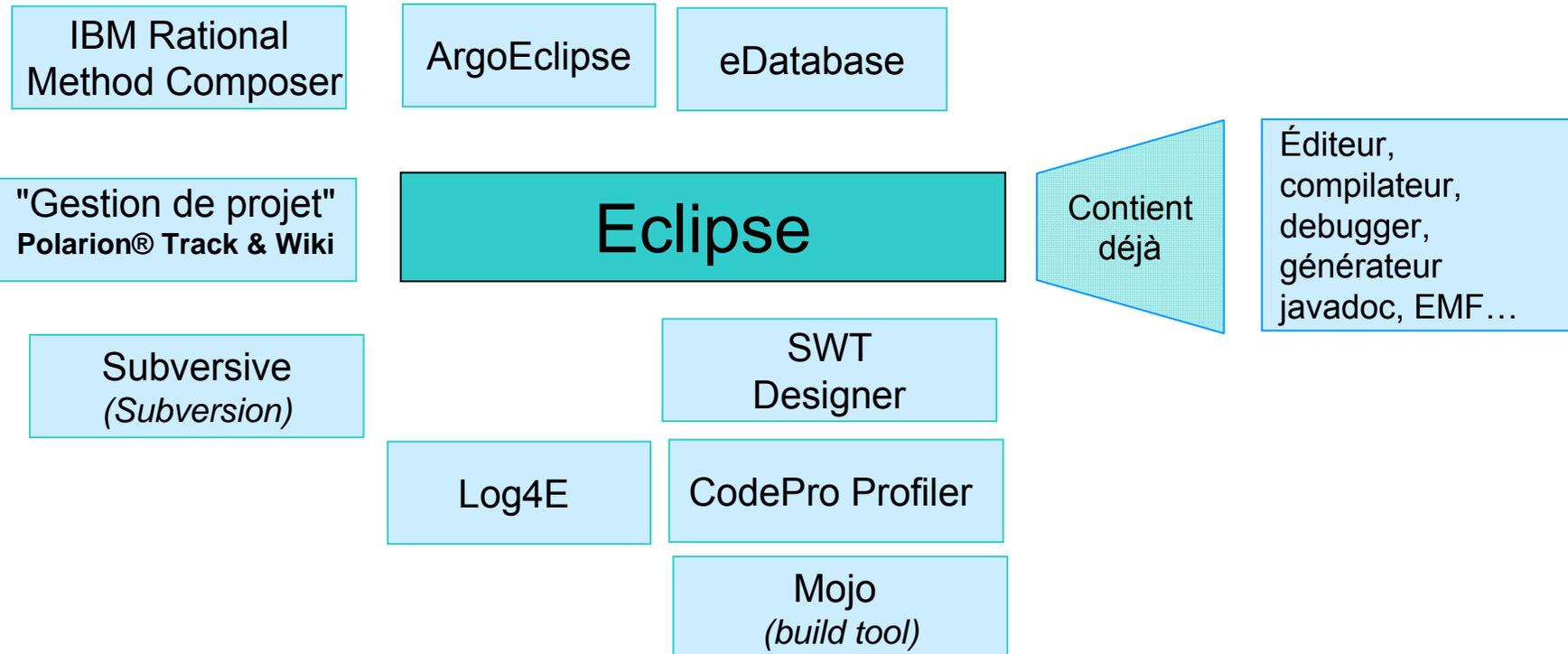
Niveaux d'assistance :

- 
- **Outils de développement** : éditeur, compilateur, debugger, profiler, gestion de version, multi-utilisateurs. *Ex : Unix*
 - **Environnements de Développement Intégré** : idem mais regroupés au sein d'une seule interface et intégrés entre eux. *Ex : Turbo C++.*
 - **Environnement de Développement Rapide** : idem avec facilité d'automatisation de certaines tâches de programmation (e.g. interfaces graphiques). *Ex : Visual x, JBuilder, NetBeans, Eclipse ...*
 - **Atelier de Génie Logiciel** : idem avec support étendu aux autres phases du cycle de développement du logiciel (spécification, conception, déploiement ...). *Ex: WinDev*

Fonctions assistant le développement

Activité	Exemples
Edition	Editeur de code avec aide maximale à l'utilisateur (complétion code, ...)
Gestion de configuration	Outils de gestion de versions (CVS), construction d'application, installateurs, ...
Développement rapide	Générateurs d'IHM, langage haut niveau, ...
Support méthodologique	Passerelle Conception/Développement
Compilation	Compilateurs, éditeurs de liens, outils de construction de code (<i>make</i> , <i>ant</i>), interpréteurs, debugger,
Programmation	Framework (Struts, JDO,...), vérification de respect de normes de codage (ex. <i>Checkstyle</i>), Dictionnaire de données, ...
Tests	Outils de benchmarking, génération / exécution / analyse de scénarii de tests, analyseur de code / optimiseur (<i>profiler</i>)... ..
Documentation	Générateurs de documentation technique, ...

Zoom sur Eclipse : configuration AGL...



Eclipse Plugin

1. Introduction

Configuration à titre indicative » basée sur une analyse fonctionnelle et non testée *intensively*

Zoom sur la gestion de configuration

Se préoccupe de la fin du cycle de développement :

- Suivre, prévoir les modifications y compris dans les autres phases (planification, conception, etc...)
- Suivre, intégrer les versions (ex: version grand public et professionnelle), et leur génération.
- Correctifs et mise à jours des anciennes versions ne se limitant pas à une désinstallation et une réinstallation.
- Suivi des versions livrées (ex: identification des logiciels concernées par une faille de sécurité sur un DLL).

=> Implique beaucoup de rigueur dans le codage et les documentations techniques ... parfois incompatible avec les aspirations créatives des développeurs.

Quelques AGL Orientés Réalisation

- **Windev de PCSoft**. Basé sur Merise (applications BD)
- **PowerBuilder de Sybase** (PowerSoft). Approche orienté objet (applications BD).
- **Oracle Developer de Oracle Corporation**. (applications BD).
- **SafeBuild de TNI-Valiosys**. Basé sur UML (développement d'applications *temps-réel*).
- **Rational Suite Development Studio** de Rational Software. Basé sur UML. Applications OO.
- **Eclipse** (<http://www.eclipse.org>)

Critères d'adoption d'un AGL 1/2

- Choisir d'utiliser un AGL pose certains questionnements :
 - **Investissement de ressources**
 - ⇒ Coût d'adoption d'une technologie AGL.
 - **Aide et Support technique disponible :**
 - ⇒ évaluation à long terme de l'exploitation du logiciel
 - **Méthodes et processus de GL existants dans l'entreprise**
 - ⇒ adéquation entre ce qui est fait par les 'acteurs' et ce qui est proposé par les outils
 - **Montée en charge**
 - ⇒ aussi bien en terme d'ampleur du projet que de la performance des applications générées avec l'outil.

Critères d'adoption d'un AGL 2/2

- **Evaluation de la valeur réelle** ⇒ écart plus ou moins grand avec les arguments commerciaux/marketing.
 - **Variété des standards** ⇒ problème de la sélection et de la comparabilité des produits.
 - **Complexité de l'adoption du produit** ⇒ en terme d'utilisation mais aussi en terme de déploiement dans l'entreprise.
- ➔ le problème du choix devient alors difficile ... ce qui explique les efforts importants de communication des éditeurs.

Bénéfices observés par les entreprises ...

- gains de productivité
- gains de qualité
- augmentation de la documentation
- amélioration de la communication
- renforcer l'utilisation de standards et de méthodologie

... a moyen terme !

... Sous réserve

Que l'entreprise prenne en compte :

- **des coûts de maintenance à long terme de l'AGL** (-> outils déjà développés)
- **Apparition fréquente de nouvelles technologies** (effet de mode ou révolution ex : C -> C++, VB -> Java)
- **Coûts de formation** (nouveaux intégrés et mise à jour/amélioration des plus anciens)
- **et les aspects classiques de l'adoption de nouvelles technologies ou outils informatiques dans l'organisation de l'entreprise** (implication de la direction, analyse de l'impact, adéquation des moyens mis en œuvre ...)
- Qu'ils correspondent à la **stratégie de l'entreprise (SSII)**: les projets courts, les contraintes de déploiement et de maintenance peuvent écarter des outils « lourds » pour des solutions logicielles au cas par cas ...

Bien évaluer le coût **réel**

- **Coût des licences et modalités** : pas de surprise sur le degré de propriété des produits développés ex: technologie propriétaire
- **Coût de la maintenance** : mises à jour et support technique (// avec prix matériel/prix consommables)
- **Coûts matériels/logiciels induits** (mise à niveau du parc matériel et/ou logiciel). Exemple plateforme de destination.
- **Coûts de formation** (logiciel/méthodologie, court/moyen terme)
- **Coûts organisationnel** : peut nécessiter une réorganisation de l'entreprise (business reengineering)
- **Coût du transfert /réutilisations d'outils/modules/librairies déjà développés**

II - Premiers Pas

- 1. Présentation générale de Windev**
- 2. Première application**

Ce que propose Windev (7.5)

Assister les phases de développement de:

- Conception (**IHM, Architecture BD, rapport technique...**),
- Implémentation (**W-Langage, IDE, RAD**),
- Test (**d'une partie ou de l'ensemble du projet**),
- Maintenance (**mise à jour produit**)
- Déploiement (**génère un installateur Windows**).

Dans l'optique de développement d'applications :

- orientée BD (**SGBD**),
- orientée réseau (**client/serveur, Internet, SGBD réseau,...**).
- en équipe de projet (**gestion multi-utilisateurs, versions**)

Plateforme de destination : Windows (3.1, 9x, noyau NT).

Nouveautés Windev 9 (sources marketing PCSoft) 1/2

Ajout :

- **Suivi de projet** : Centre de Contrôle (début de gestion de ressources, production de tableau de bord)...
- **Optimiseur code** (SQL, code RAD)
- **Génération du code d'une application en Java** (Développement Windev -> Implantation Java) (non testée).
- **Acces BD format XML.**
- **Gestion de configuration** (1 projet pour plusieurs version d'un logiciel)

Amélioration:

- **Simplification du code**
- **Génération automatique de code** (rRAD pour real RAD, personnalisation des interfaces,)
- **Ergonomie et librairie** (petites astuces d'IHM, modèles disponibles, ...)

Nouveautés Windev 9 (sources marketing PCSoft) 2/2

Dans l'optique de nouveaux type d'applications ...

- Logiciels embarqués Mobiles (Pocket PC)

... sur d'autres plateformes de destination :

- Mobiles (Pocket PC)
- Linux ou autres (génération de code java, BD et Serveur Hyperfile portés sous Linux)

Zoom sur la phase de conception

■ Méthodologie:

- **Support limité UML** : diagramme utilisation & diagramme de classe

■ IHM :

- **boîte à outils de composants graphiques**
- **correcteur d'interfaces graphique** (esthétique plutôt qu'ergonomique)
- **étude enchaînement des fenêtres** (a posteriori)

■ BD

- **Support limité MERISE** : Éditeur de MCD, MLD

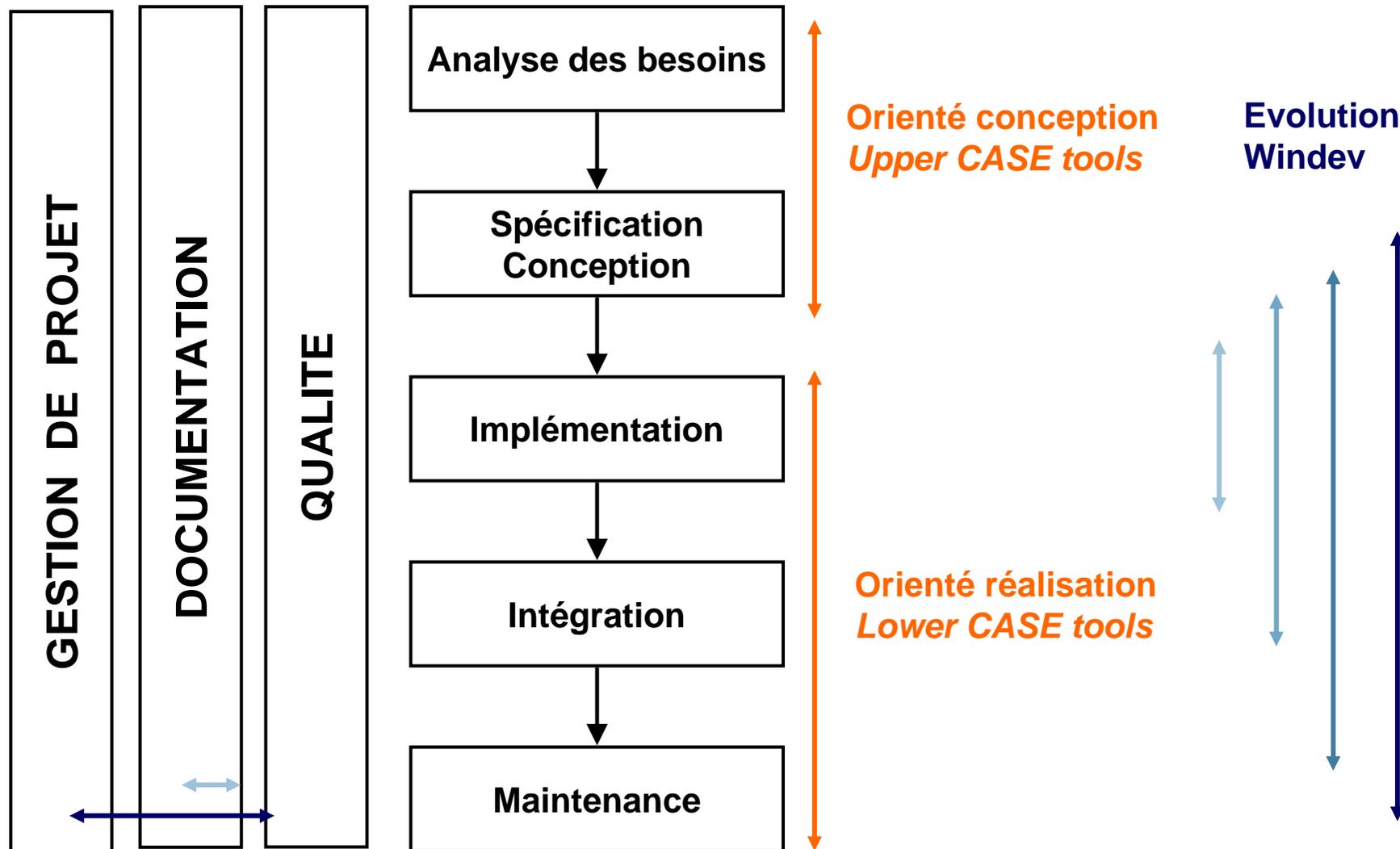
■ Documentation technique :

- **Génération automatique de la documentation technique** (dictionnaire données, code source, captures GUI)

Zoom sur la phase de développement

- Langage dédié : **W Langage**
- Dédié à des projets de taille moyenne (**1 à 50 personnes**)
- Ouvert sur d'autres langages (**Java, C++, C#, VB, XML**)
- Editeur et débbugger intégrés
- RAD pour le développement orienté BD
- Gestion de version
- Gestion de l'intégration (**installation, modules, système cible 16/32, tests modulaires**)
- Gestion de mise à jour

Positionnement de Windev ./ GL & AGL



2. 1er pas avec Windev

Création d'un projet Windev

Un projet Windev est un ensemble de fichiers (code, conception GUI, BD,...) aboutissant au développement d'une application.

Deux grands types d'applications se dégagent :

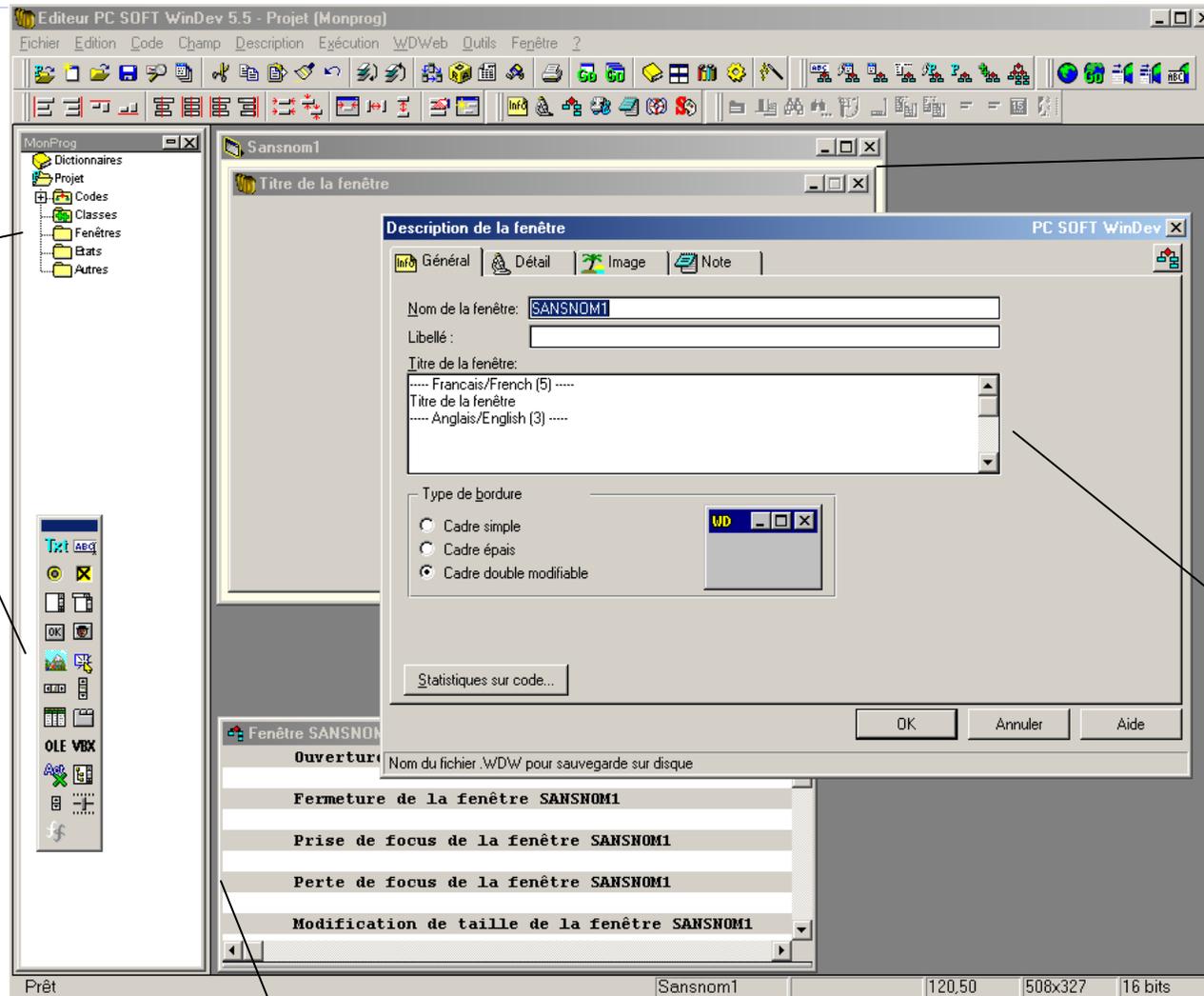
- Applications graphiques Windows.
- Applications Gestion de Bases de Données.

Chronologie de la création d'une application graphique : (s) signifie "cliquer sur Suivant".

1. **Fichier>Projet>Nouveau...(s)**
2. **Nommer le projet et sa localisation(s).**
3. **Vérifiez qu'aucun format de BD n'est sélectionné.**
4. **Créer une feuille de style (s), sélectionner un style (s).**

2. 1er pas avec Windev

L'éditeur de fenêtre Windev 5.5



Fenêtre

Composant graphique en cours de conception

Propriété

Edition des propriété de la fenêtre.

TreeView

Liste des objets composant le projet.

Champs

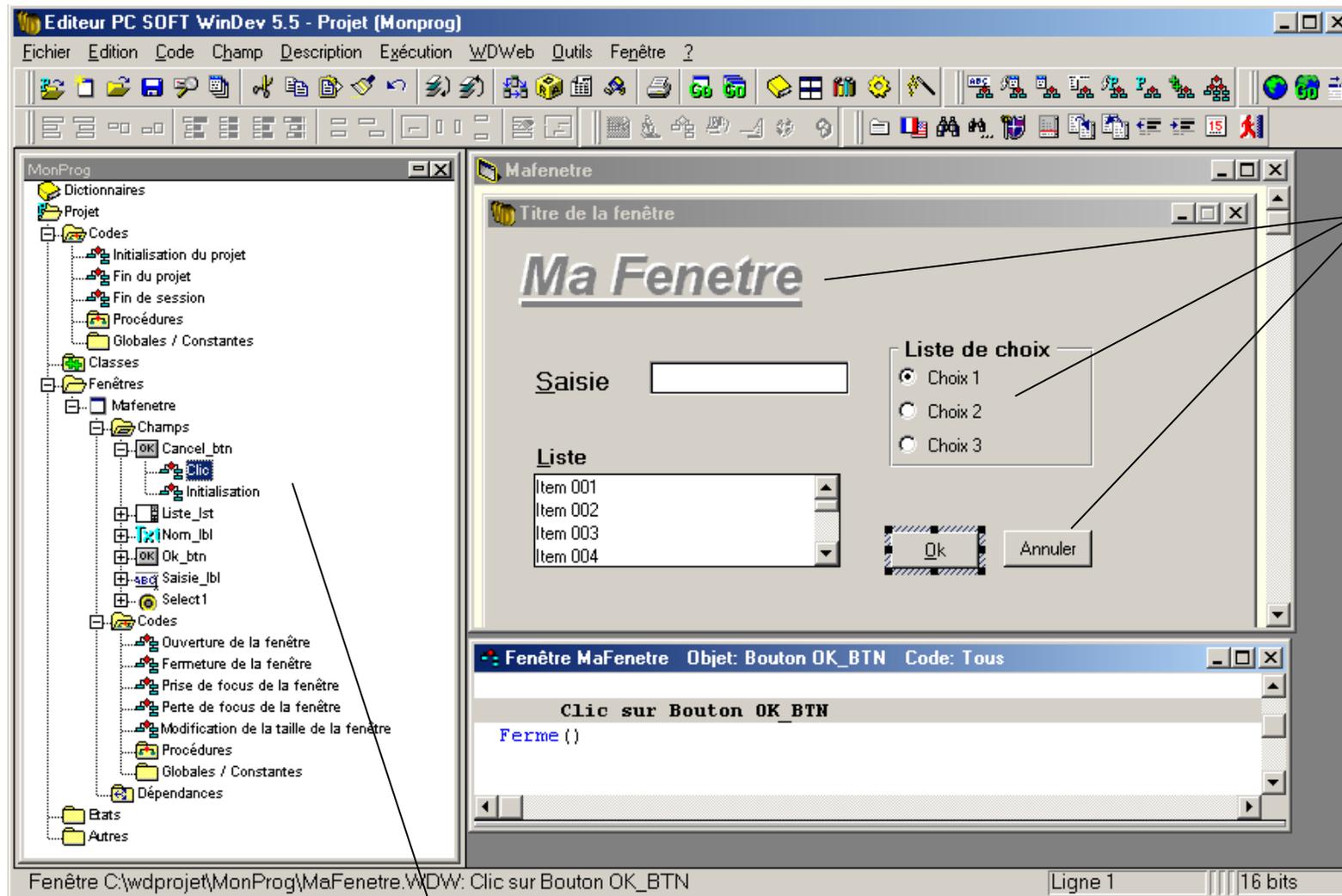
Objets graphiques pouvant être inséré dans la fenêtre (Bouton, zone de texte, ...)

Code

Listes des événements gérés par le composant graphique et leur comportement associé défini avec le W Langage.

2. 1er pas avec Windev

Exemple de conception de fenêtre (Windev 5.5)



Chaque composant graphique est identifié par un nom dont la portée est celle de la fenêtre dans laquelle il s'inscrit.

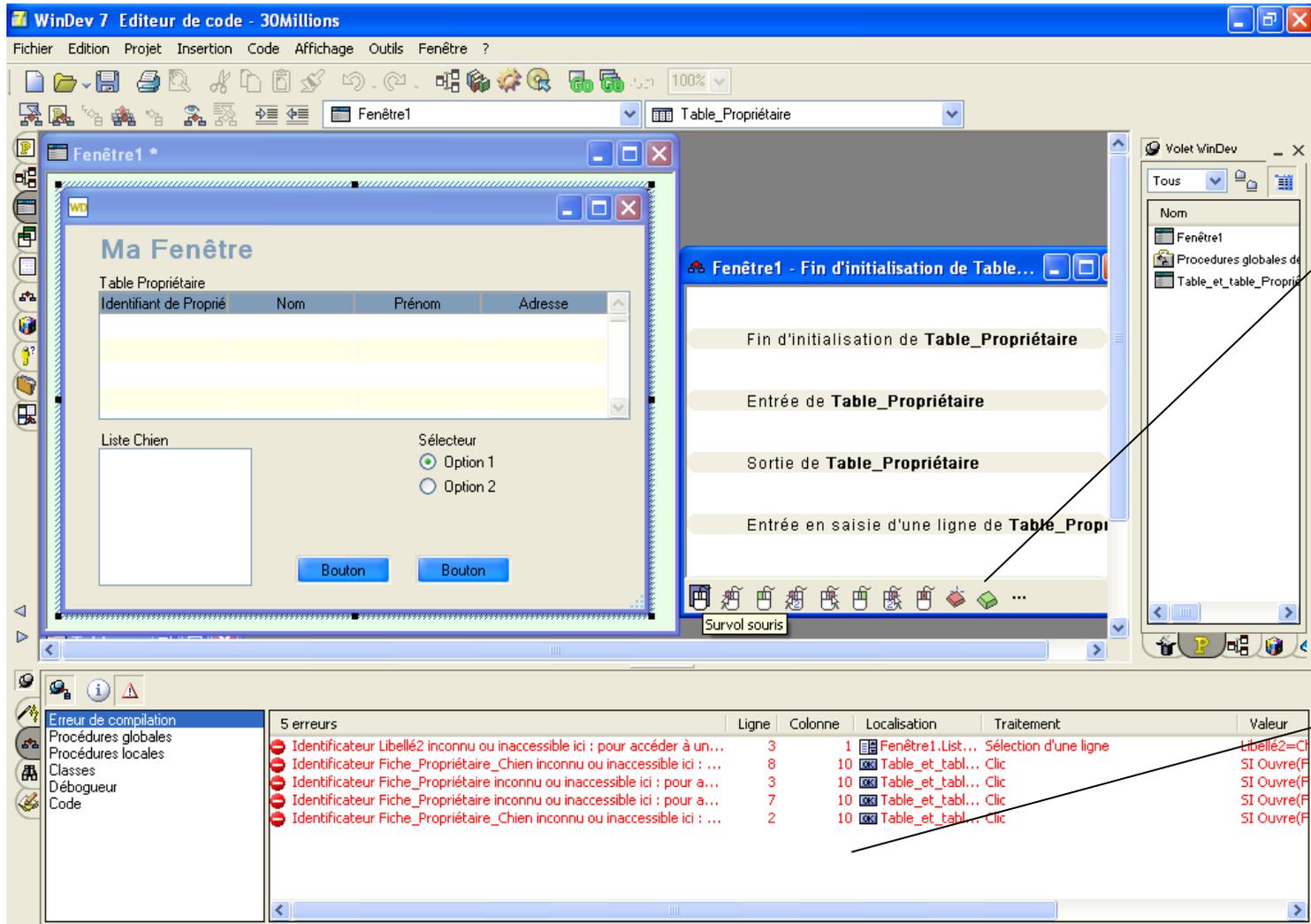
Les propriétés du composant seront accédées à partir de ce nom. Ex :

Label..Valeur :
contient la valeur saisie dans le champs de saisie.
Label..Libellé :
contient le texte du libellé.

2. 1er pas avec Windev

Chaque composant listé dans le **treeview** est décrit par son type par une icône (ex : bouton) et les événements gérés (et le code associé).

Exemple de conception de fenêtre (Windev 7.x)



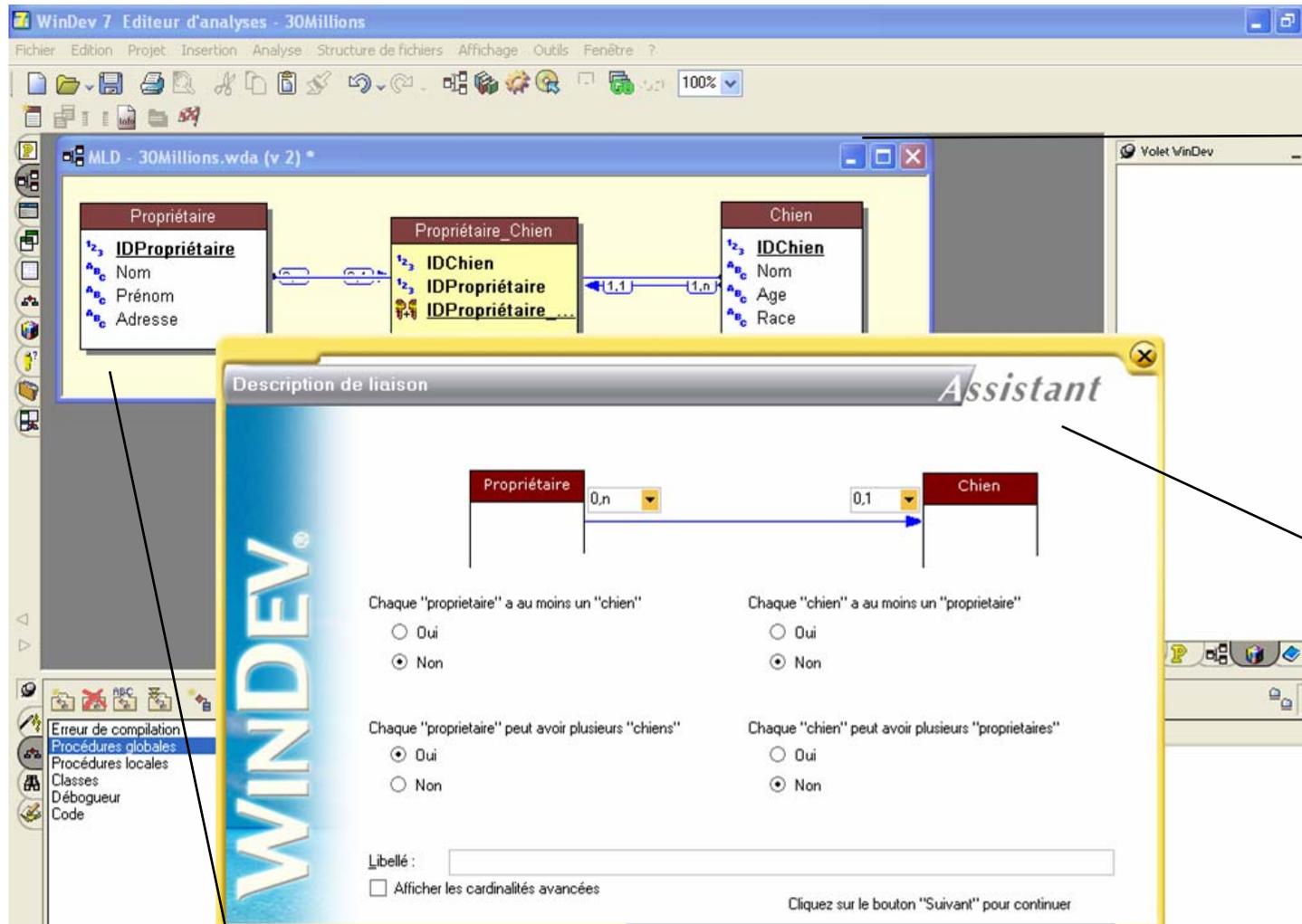
Nouveaux événements

Remplace le TreeView

2. 1er pas avec Windev

Le **kouglov** liste les erreurs de programme, procédures locales et globales, classes, appel débogueur, liste de tâches.

L'éditeur d'analyse



MLD (Modèle Logique de Donnée)

Modèle Entité-Association décrivant la structure de la BD.

Assistant Merise

Avec des questions simples, aide à établir les cardinalités des associations entre entités.

2. 1er pas avec Windev

Entité

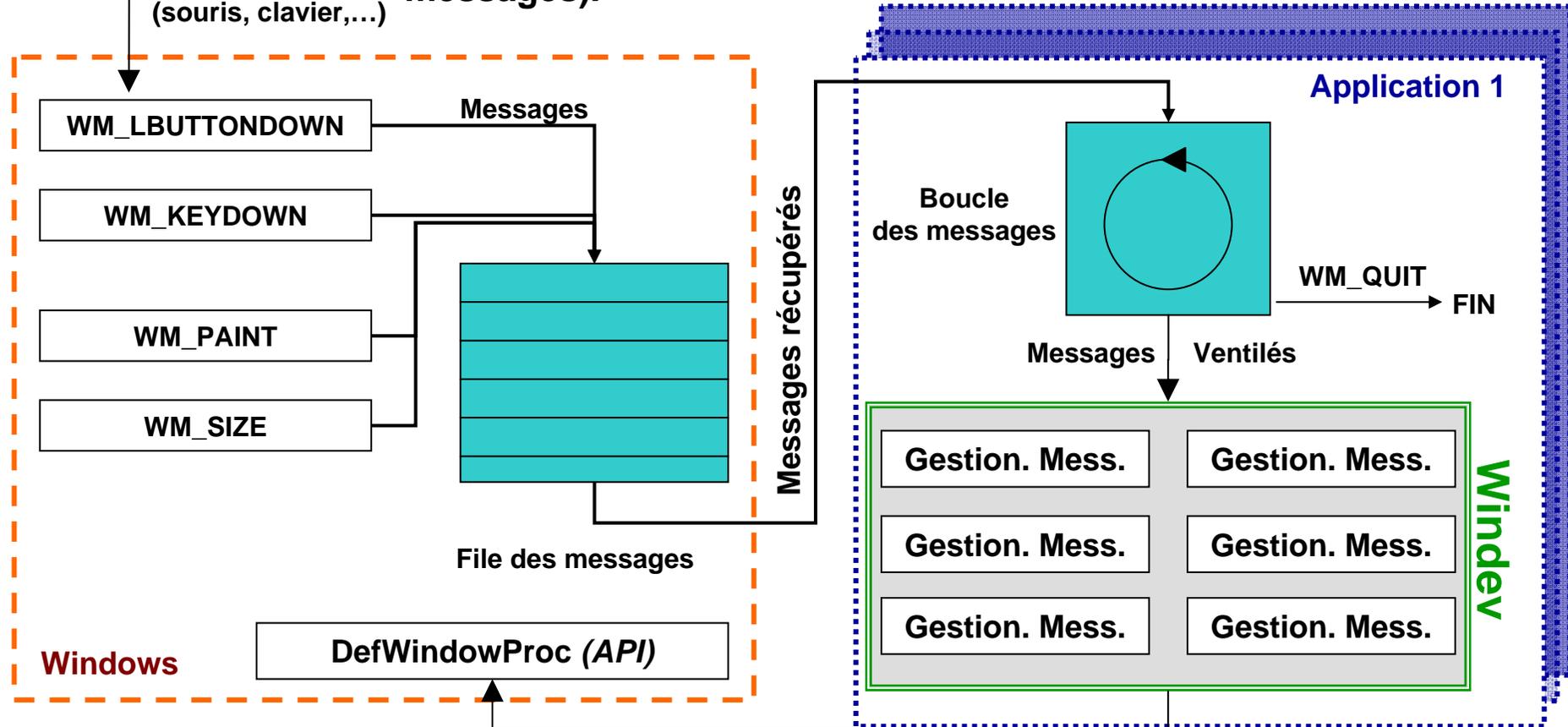
Les propriétés des entités sont établies à l'aide d'une autre éditeur permettant de préciser le type, le format etc.. des propriétés.

Principe de la programmation graphique sous Windows



Programmation événementielle : l'application réagit aux événements générés par l'utilisateur via le clavier, la souris ... (encore appelés messages).

Evenements (souris, clavier,...)



2. 1er pas avec Windev

Messages non traités

Première Application

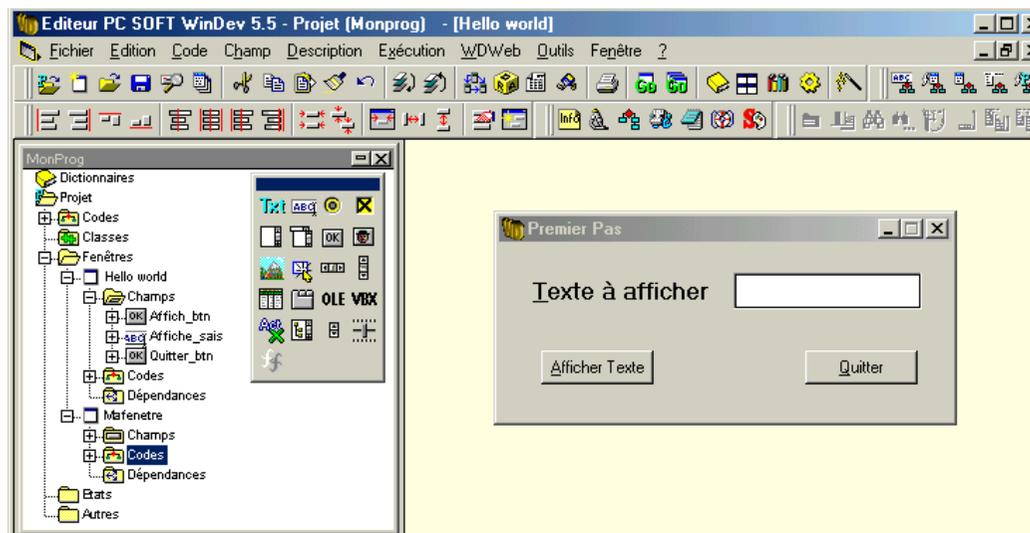
- Nous allons créer une application qui affiche dans une fenêtre "popup" un texte saisi par l'utilisateur.

Composants utilisés :

- Fenêtre
- Bouton
- Champs de Saisie

Composants prédéfinis utilisés :

- Fenêtre d'information (fenêtre + 1 bouton Ok)
- Fenêtre de confirmation (fenêtre + 2 boutons Oui, Non)



Nous verrons quelques exemple de code W-Langage utilisé pour définir le comportant des composants graphiques

2. 1er pas avec Windev

Le champ de saisie Windev 5.5

Permet à l'utilisateur de transmettre des informations à l'application

Description d'un champ de saisie PC SOFT WinDev

Général Détail Fichier Note Aide Style

Nom: AFFICHE_SAIS

Libellé:

- Francais/French (5) ----
- &Texte à afficher
- Anglais/English (3) ----

Masque de saisie/affichage:
(Déroulez la liste ou saisissez le masque)

<aucun> Taille maxi de saisie: 20

Type du champ

- Texte
- Numérique
- Heure
- Date
- Monétaire WinDev
- Monétaire WinDev+Euro

Gestion de l'EURO

Devise mémorisée: <défaut> Arrondir à: 2 décimales

Visible dans la bulle d'aide:

Devise de saisie/d'affichage: <défaut> Modifiable par menu contextuel:

Afficher ~ en cas d'écart d'arrondi:

Double: <aucun> Cette devise est celle utilisée à la saisie et à l'affichage

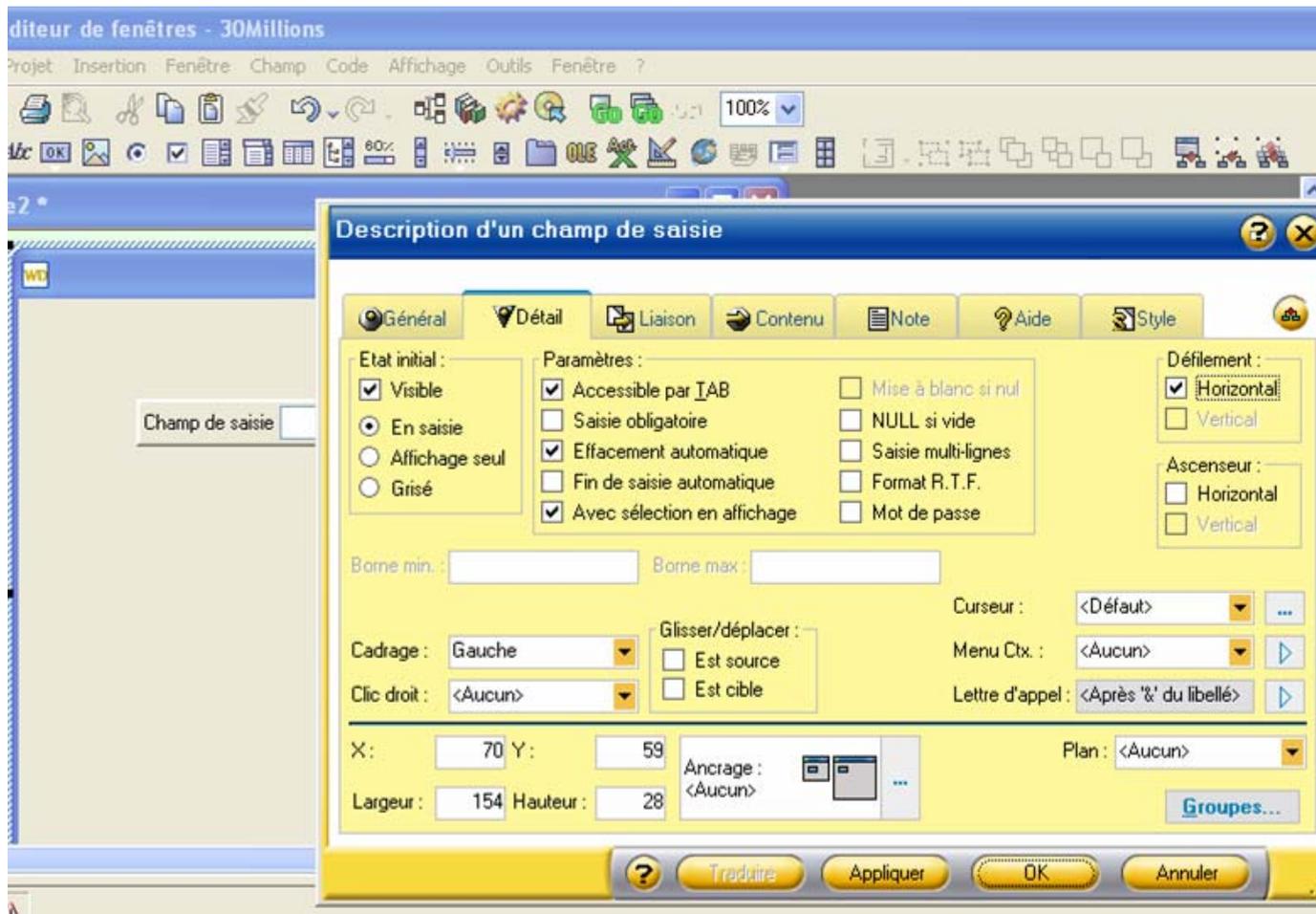
OK Annuler Aide

Libellé affiché à côté du champ

2. 1er pas avec Windev

Le champ de saisie Windev 7.x

Propriétés supplémentaires ...



2. 1er pas avec
Windev

Le champ Bouton d'affichage

Le bouton "Affich_Btn" permet d'ouvrir une fenêtre d'information contenant le texte saisi dans le champ de saisie.

Code correspondant : `info("<<" + AFFICHE_SAIS..Valeur + ">>")`

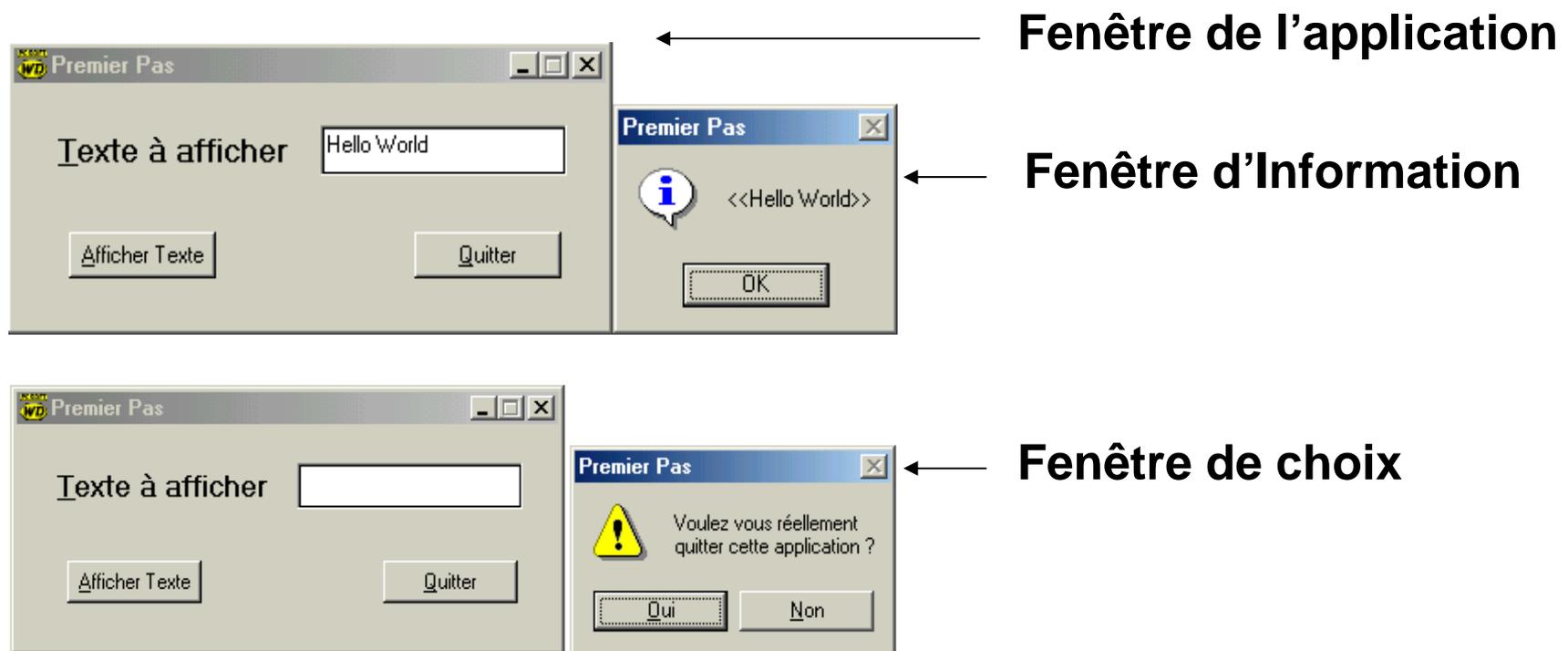
(1 seul événement)

2. 1er pas avec Windev

Résultat de l'exécution

Le bon fonctionnement de la fenêtre peut être testée avant génération de l'exécutable à l'aide du bouton 

Le bouton  permet de tester le projet complet.



2. 1er pas avec Windev

- http://www.adeli.org/wiki/?title=Normes%2COutils_en_GCL