

## Processus Unifié – Unified Process

### Introduction

Un processus de développement définit une séquence d'étapes, en partie ordonnée, qui concourt à l'obtention d'un système logiciel ou à l'évolution d'un système existant ; pour produire des logiciels de qualité, qui répondent aux besoins des utilisateurs dans des temps et des coûts prévisibles.

Architecture, Pattern, Savoir-faire, sont des aspects qui voient leur importance dans le processus de développement bien établie. Cependant, les modèles de conception (design patterns) ne font que commencer à être intégrés dans les outils, et d'une manière rarement normative. Il en est de même pour la prise en compte des architectures logicielles. L'explicitation de ces informations sous forme de méta-modèles standard constitue un préliminaire à leur opérationnalité et à la généralisation de leur utilisation.

## 1 Processus de développement 2TUP

### 1.1 les processus unifiés

Un processus unifié est un processus construit sur UML (Unified Modeling Language). Les processus unifiés sont le résultat de l'unification, non pas des processus, mais plus exactement les meilleures pratiques du développement objet.

Un processus unifié se distingue par les caractéristiques suivantes :

- **Itératif** : Le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs et développer une solution effective de façon incrémentale par des itérations multiples.
- **Piloté par les risques** : les causes majeures d'échec d'un projet logiciel doivent être écartées en priorité.
- **Centré sur l'architecture** : le choix de l'architecture logicielle est effectué lors des premières phases de développement du logiciel. La conception des composants du produits est basée sur ce choix.
- **Conduit par les cas d'utilisation** : le processus est orienté par les besoins utilisateurs présentés par des cas d'utilisation.

Le tableau suivant présente une synthèse des processus les plus en vogue dans la communauté objet et nouvelles technologies.

## Free Information Systems For Management Application

Les processus unifiés sont présentés par les processus RUP et 2TUP :

	Description	Points forts	Points faibles
<b>Cascade</b>	<ul style="list-style-type: none"> <li>- Propose de dérouler les phases projet de manière séquentielle</li> <li>- Cité pour des raisons historiques</li> </ul>	<ul style="list-style-type: none"> <li>- Distingue clairement les phases projet</li> </ul>	<ul style="list-style-type: none"> <li>- Non itératif</li> <li>- Ne propose pas de modèles de documents</li> </ul>
<b>RUP Rational Unified Process</b>	<ul style="list-style-type: none"> <li>- Promu par Rational.</li> <li>- Le RUP est à la fois une méthodologie et un outil prêt à l'emploi (documents types partagés dans un référentiel Web)</li> <li>- Cible des projets de plus de 10 personnes</li> </ul>	<ul style="list-style-type: none"> <li>- Itératif</li> <li>- Spécifie le dialogue entre les différents intervenants du projet : les livrables, les plannings, les prototypes...</li> <li>- Propose des modèles de documents, et des canevas pour des projets types</li> </ul>	<ul style="list-style-type: none"> <li>- Coûteux à personnaliser</li> <li>- Très axé processus, au détriment du développement : peu de place pour le code et la technologie</li> </ul>
<b>XP eXtreme Programming</b>	<ul style="list-style-type: none"> <li>- Ensemble de « Bests Practices » de développement (travail en équipes, transfert de compétences...)</li> <li>- Cible des projets de moins de 10 personnes</li> </ul>	<ul style="list-style-type: none"> <li>- Itératif</li> <li>- Simple à mettre en œuvre</li> <li>- Fait une large place aux aspects techniques : prototypes, règles de développement, tests...</li> </ul>	<ul style="list-style-type: none"> <li>- Ne couvre pas les phases en amont et en aval au développement : capture des besoins, support, maintenance, tests d'intégration...</li> <li>- Elude la phase d'analyse, si bien qu'on peut dépenser son énergie à faire et défaire</li> <li>- Assez flou dans sa mise en œuvre: quels intervenants, quels livrables ?</li> </ul>
<b>2TUP Two Track Unified Process</b>	<ul style="list-style-type: none"> <li>- S'articule autour de l'architecture</li> <li>- Propose un cycle de développement en Y</li> <li>- Détaillé dans "UML en action" (voir références)</li> <li>- Cible des projets de toutes tailles</li> </ul>	<ul style="list-style-type: none"> <li>- Itératif</li> <li>- Fait une large place à la technologie et à la gestion du risque</li> <li>- Définit les profils des intervenants, les livrables, les plannings, les prototypes</li> </ul>	<ul style="list-style-type: none"> <li>- Plutôt superficiel sur les phases situées en amont et en aval du développement : capture des besoins, support, maintenance, gestion du changement...</li> <li>- Ne propose pas de documents types</li> </ul>

*Synthèse des méthodologies utilisées dans le cadre de développement Objet et nouvelles technologies.*

### 1.2 Le processus 2TUP

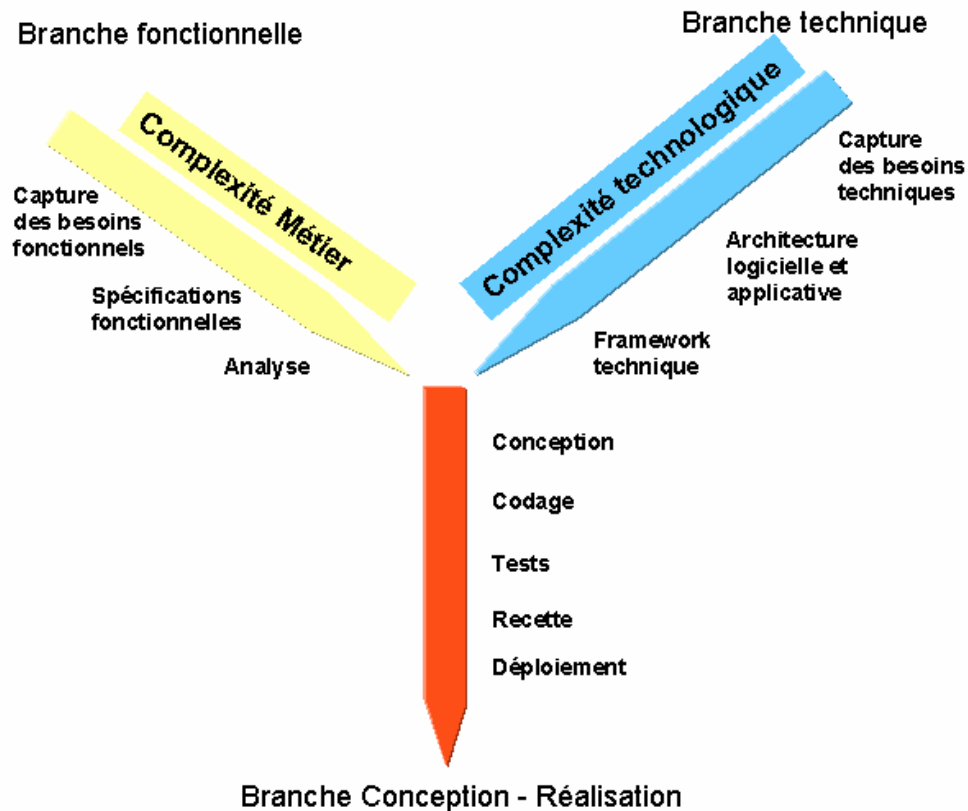
Le processus 2TUP (Two Track Unified Process) est un processus unifié. Il gère la complexité technologique en donnant part à la technologie dans son processus de développement.

Le 2TUP propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels et propose une étude parallèle des

deux branches : fonctionnelle (étude de l'application) et la technique (étude de l'implémentation). Illustré sur la figure suivante, le processus 2TUP s'articule autour de trois phases :

- Une branche technique
- Une branche fonctionnelle
- Et une branche de conception réalisation.

La figure suivante détaille les étapes de développement des trois branches du processus 2TUP.



***Le processus 2TUP***

### **1.2.1 Branche fonctionnelle**

Les principales étapes de la branche fonctionnelle se présentent comme suit :

L'étape capture des besoins fonctionnels produit le modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système inadapté aux utilisateurs. Cette phase a pour objectif de définir :

- La frontière fonctionnelle entre le système considéré comme une boîte noire et son environnement, c'est le niveau contexte.
- Les activités attendues des différents utilisateurs par rapport au système toujours envisagé comme une boîte noire, c'est le niveau cas d'utilisation.

L'étape d'analyse consiste à étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

### **1.2.2 Branche technique**

Les principales étapes de la branche technique se présentent comme suit : L'étape capture des besoins techniques recense toutes les contraintes sur les choix de dimensionnement et la conception du système. Les outils et le matériel sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique). Cette étape permet de définir le modèle d'analyse technique. Le rôle de ce dernier est d'établir les couches logicielles et y spécifie les activités techniques attendues.

L'étape conception générique définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle permet de générer le modèle de conception technique ou design pattern (aspect qui sera développé ultérieurement) qui définit les Frameworks. Ces derniers, délivrant les services techniques, assurent la réponse aux exigences opérationnelles du système.

### **1.2.3 Branche conception - réalisation**

Les principales étapes de cette branche se présentent comme suit : L'étape conception préliminaire est une étape délicate, car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer. Cette étape permet de produire le modèle de conception système. Ce dernier organise le système en composants, délivrant les services techniques et fonctionnels. Ce modèle regroupe les informations des branches technique et fonctionnelle.

L'étape conception détaillée permet d'étudier comment réaliser chaque composant. Cette étape produit le modèle de conception des composants. Ce modèle fournit l'image prête à fabriquer du système complet.

C'est dans l'étape de codage que s'effectue la production des composants et les testes des unités de code au fur et à mesure de leur réalisation.

L'étape de recette consiste à valider les fonctionnalités du système développé.