

Le processus unifié (UP) est une méthode générique de développement de logiciel. Son objectif est de maîtriser la complexité des projets informatiques en diminuant les risques.

Les principes de UP :

- **Utilise le langage UML.**
- **guidé par les cas d'utilisation** : décrit les interactions entre acteurs et système , capturer les vrais besoins et favorise le développement itératif
- **itératif et incrémental** : itérations successives, chaque version est testée et reçue par l'utilisateur
- **centré sur l'architecture** : mettre en avant l'architecture dès le début des travaux d'analyse et conception. Le bon choix de l'architecture facilite l'implémentation et le déploiement.
- **réduction des risques** : minimisation des risques au fur et à mesure des itérations

On distingue principalement deux type de processus unifiés : **RUP** et **2TUP**

Rational Unified Process (RUP) :

Principe : il se compose de **4 phases** , **5 activités** et un **ensemble d'itérations**

4 phases : Etude d'opportunité / Élaboration / Construction / Transition

5 activités : Expression des besoins / Analyse / Conception / Implémentation / Test

Ensemble d'itérations : circuit de développement aboutissant à un livrable.

2 Track Unified Process (2TUP) :

Objectif : prendre en compte les contraintes de changement imposées aux systèmes d'information.

Principe : Le processus suit deux chemins : **Fonctionnel** et **Technique**

Caractéristiques : itératif et incrémental piloté par les risques / piloté par les exigences utilisateurs / modélisé avec UML / centré sur l'architecture

Branche fonctionnelle : **capture des besoins fonctionnels + analyse**

Branche technique : **capture des besoins techniques + conception générale**

Par la suite , les deux branches se fusionne en une seule branche qui comporte les phases restantes : **Conception préliminaire + Conception détaillée + Codage et tests + Recette**

Inconvénients de UP :

- Fait tout , mais lourd.
- Parfois difficile à mettre en œuvre de façon spécifique.

UP est pour les grands projets qui génèrent beaucoup de documentation

D'où la nécessité des méthodes agiles (eXtrem Programming / SCRUM / ...)