

P.3

Définir le processus unifié (UP)

Objectifs

Le processus unifié fournit un cadre au développement logiciel pour la construction de systèmes orientés objet.

Le processus unifié répond aux exigences fondamentales suivantes :

- être guidée par les besoins des utilisateurs
- être centrée sur l'architecture logicielle
- être itérative et incrémentale

Préambule

Il ne s'agit pas d'un cycle en cascade séquentiel. Le cycle de développement se compose de nombreuses **itérations**.

Les phases

Les phases d'un processus de développement sont des états de celui-ci à un instant t. Le cycle de développement du Processus Unifié organise les tâches et les itérations en quatre phases :

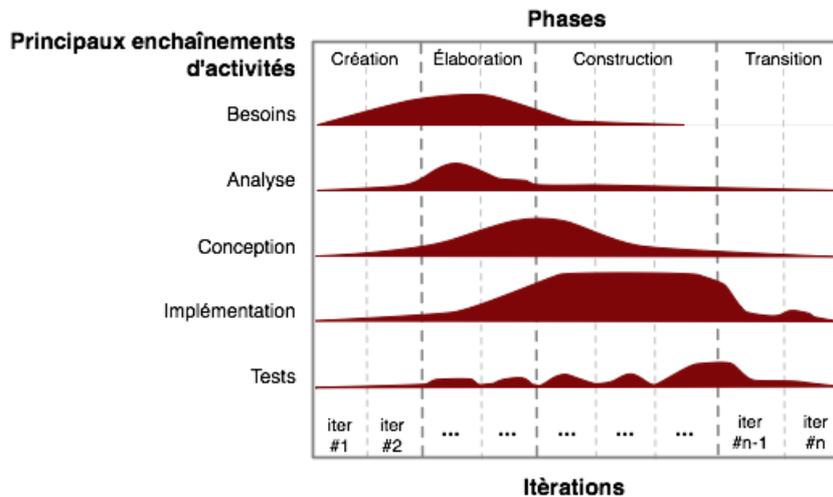
- Inception : spécification des besoins et aussi une sorte d'étude faisabilité où on effectue les recherches nécessaires pour décider si on poursuit ou non le projet
- Élaboration : on développe de façon incrémentale l'architecture du noyau, les risques et la plupart des besoins sont identifiés
- Construction : on construit des sous-ensembles exécutables et stables du produit final
- Transition : le produit final est livré en version bêta à la disposition des utilisateurs

Les activités

Les activités représentent les actions à effectuer au cours d'une phase : une phase passe par l'ensemble des activités. Le temps passé par activité est fonction des phases.

Les activités de développement sont définies par des disciplines ou workflows fondamentales suivantes : Modélisation métiers, Exigences, Conception, Implémentation, Tests, Déploiement, Gestion de la configuration, Gestion de projet, Environnement.

La répartition de l'effort change avec le temps. Les premières itérations ont tendance à mettre l'accent sur certaines activités (analyse et conception), les autres itérations moins. Les efforts seront ensuite portés sur l'implémentation, les tests et le déploiement.



<i>Modélisation de métier</i>	Visualise les concepts du domaine les plus importants	Modèle du domaine diagramme de classes métiers
<i>Exigences</i>	Compréhension et expression des besoins et des exigences du client.	Modèle d'analyse diagramme des cas d'utilisation spécifications supplémentaires glossaire scénarios prototypage de l'IHM plan de tests de validation
<i>Conception</i>	Décrit les différentes vues (fonctionnelles, dynamique et statique) d'une architecture.	diagramme de classes de conception diagramme d'interaction (séquence) diagramme d'activités et/ou d'états diagramme de déploiement plan de tests d'intégration
<i>Implémentation</i>	C'est le résultat de la conception pour implémenter le système sous formes de composants On planifie les intégrations des composants pour chaque itération, On produit les classes et les sous-systèmes sous formes de codes sources.	code source, scripts, binaires, exécutables et autres éléments du même type (tables du base de données). respect des standards de codage diagramme de composant diagramme de déploiement plan de tests unitaires
<i>Test</i>	Les tests permettent de vérifier des résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.	rapport des tests unitaires, d'intégration et de validation
<i>Déploiement</i>	Couvre la configuration du système à livrer.	manuels d'installation et d'utilisation
<i>Gestion de la configuration</i>	Contrôle les modifications du système.	système de gestion de version (subversion)
<i>Gestion de projet</i>	Décrit différentes stratégies de travail avec un processus itératif.	diagramme de gantt répartition des tâches
<i>Environnement</i>	Couvre l'infrastructure nécessaire demandée pour développer.	ressources opératives, logicielles, matérielles et de documentation