

Processus de Développement Logiciel - PDL

Leonard Cseres | June 12, 2025

I. Processus de Développement Logiciel

- **Définition:** Décomposition en activités structurées qui mènent à la production de logiciel. Représentation abstraite d'un processus.
- **Activités fondamentales:**
 - **Spécification:** Définir ce que le logiciel doit faire. Formulation des besoins et exigences. Compréhension des attentes des parties prenantes.
 - **Conception:** Établir la structure du système. Conversion de la spécification en description de structure. Inclut la conception architecturale, des bases de données, des interfaces et des composants.
 - **Implémentation:** Écriture du code. Peut être tactique (rapidité) ou stratégique (structure à long terme).
 - **Vérification et Validation:**
 - * **Vérification:** "Est-ce que je construis le système correctement?" Conformité aux spécifications, tests.
 - * **Validation:** "Est-ce que je construis le bon système?" Répond aux besoins utilisateurs.
 - **Évolution:** Modification du système en réponse à de nouveaux besoins.
- **Décomposition de problèmes:** Division des problèmes complexes en parties indépendamment résolubles.
- **Modèles de processus:**
 - **Modèles en cascade:** Processus planifié à l'avance; phases distinctes.
 - * **Caractéristiques:** Phases séparées et indépendantes; pas d'adaptation pendant l'exécution.
 - * **Avantages:** Approprié pour grands projets, exigences stables, équipes distribuées.
 - * **Inconvénients:** Difficile à mettre à jour, manque de flexibilité. Exemples: Waterfall, V-cycle, Spiral.
 - **Modèles incrémentaux:** Planification incrémentale; processus adapté aux besoins clients.
 - * **Caractéristiques:** Spécification, conception, développement et validation entrelacés.
 - * **Avantages:** Tests et validation intermédiaires, coûts d'adaptation réduits.
 - * **Inconvénients:** Communication cruciale, dégradation de l'architecture. Exemples: RAD, RUP, XP.
 - **Modèles d'intégration et de configuration:** Réutilisation et configuration de composants existants.
 - * **Types de logiciels réutilisables:** Applications configurables, bibliothèques, services web.
 - * **Avantages:** Livraison rapide, coûts et risques réduits.
 - * **Inconvénients:** Perte de contrôle, compromis entre besoins et fonctionnalités.
- **Intégration des changements:**
 - **Prototype:** Version initiale pour prouver la faisabilité. Ne jamais mettre en production.

- **Développement par incrément:** Logiciel développé et livré par incréments.
- **Amélioration du processus:**
 - **Niveaux de maturité:** Initial, Géré, Défini, Quantitativement géré, Optimisant.
 - **Cycles d'amélioration:** Mesurer, Analyser, Changer.

II. Ingénierie des Exigences

- **Définition:** Établir les services requis par le client et les contraintes du système. Spécification de ce qui doit être implémenté.
- **Niveaux d'abstraction:**
 - **Exigences métier:** Compréhensibles par les décideurs. "Pourquoi?"
 - **Exigences utilisateur:** Langage naturel pour utilisateurs sans connaissances en informatique. "Quoi?" (haut niveau)
 - **Exigences système:** Descriptions techniques détaillées. Base contractuelle pour l'implémentation.
- **Catégories d'exigences:**
 - **Fonctionnelles:** Services fournis par le système, réactions aux entrées et situations.
 - **Non-fonctionnelles:** Contraintes sur les fonctions du système. Idéalement quantitatives pour permettre des tests.
 - **Liées au domaine:** Lois sur la vie privée, confidentialité.
- **Processus d'ingénierie des exigences:** Élicitation, Spécification, Validation.
 - **Élicitation des exigences:** Découverte et compréhension des exigences.
 - * **Techniques:**
 - **Entretiens:** L'analyste pose des questions pour comprendre les besoins utilisateurs.
 - **Approche ethnographique:** Étude descriptive et analytique.
 - **Histoires et scénarios:** Les utilisateurs décrivent comment ils gèrent les situations.
 - **Spécification:** Documentation des exigences adaptée au public.
 - * **Langages:** Naturel, naturel structuré, notations graphiques (UML), méthodes formelles.
 - * **Langage naturel:** Phrases numérotées, langage standardisé, sans jargon informatique.
 - * **Langage naturel structuré:** Utilisation de modèles avec champs prédéfinis.
 - * **Use case (UML):** Scénarios représentables, identification des acteurs et interactions.
 - **Validation de la spécification:** Propriétés: Valide, Cohérent, Complet, Réaliste, Vérifiable, Compréhensible, Traçable.

UML

- **Diagramme de cas d'utilisation:**
 - **Objectif:** Montre les interactions entre un système et son environnement.
 - **Éléments:**
 - * **Acteurs:** Entités interagissant avec le système.

- * **Cas d'utilisation:** Interactions ou services fournis aux acteurs.
- * **Relations:** include (étapes obligatoires), extend (étapes optionnelles).

- **Diagramme d'activité:**
 - **Objectif:** Décrit le flux d'activités dans un processus.
 - **Éléments:**
 - * **Nœuds de début et fin**
 - * **Actions:** Étapes individuelles.
 - * **Décisions et fusions:** Contrôle du flux.
 - * **Fourches et jonctions:** Exécution parallèle.
 - * **Couloirs:** Indiquent la responsabilité.
- **Diagramme de séquence:**
 - **Objectif:** Modélise les interactions entre objets dans un scénario.
 - **Éléments:**
 - * **Participants:** Objets impliqués dans l'interaction.
 - * **Lignes de vie:** Existence d'un participant dans le temps.
 - * **Messages:** Communications entre participants.
 - * **Spécifications d'exécution:** Activité d'un participant.
 - * **Cadres:** Constructions de flux comme opt, alt, loop.
- **Diagramme de classes:**
 - **Objectif:** Structure du système, classes et leurs relations.
 - **Éléments:**
 - * **Classes:** Catégories d'objets avec attributs.
 - * **Associations:** Relations entre classes, avec multiplicité.
 - * **Héritage:** Relations hiérarchiques.
- **Diagramme d'états:**
 - **Objectif:** Illustre le cycle de vie d'un objet.
 - **Éléments:**
 - * **États:** Conditions d'un objet.
 - * **États initiaux et finaux**
 - * **Transitions:** Déplacement entre états, déclenchées par événements.
 - * **Actions:** Activités lors de l'entrée, sortie ou pendant un état.
 - * **États imbriqués et états d'historique**