



# Activité 3.1 Modèles de cycle de vie

Modèle de cycle de vie	Caractéristiques	Avantages	Inconvénients	Caractéristiques du projet adéquat
<b>Cascade</b>	Processus linéaire et séquentiel, chaque phase commence uniquement lorsque la précédente est terminée.	- Simple à comprendre et à gérer. - Documentation claire. - Bon pour les projets bien définis.	- Rigidité, pas de retour en arrière possible. - Les erreurs sont détectées tardivement. - Long cycle de développement avant d'obtenir un produit fonctionnel.	Convient aux projets <b>stables</b> , bien définis dès le départ, avec peu d'incertitude et peu de changements (ex: systèmes critiques, logiciels de gestion).
<b>V</b>	Variante du modèle en cascade, chaque phase de développement a une phase de validation correspondante (tests).	- Meilleur contrôle de qualité grâce aux tests intégrés à chaque étape. - Facilité de détection et correction des erreurs.	- Toujours rigide, peu flexible aux changements en cours de développement. - Nécessite une planification détaillée dès le départ.	Projets <b>critiques</b> nécessitant une validation rigoureuse (ex: systèmes embarqués, logiciels médicaux, aviation).
<b>Prototypage</b>	Développement rapide d'un prototype	- Meilleure compréhension des besoins	- Peut augmenter les coûts si de	Projets <b>avec beaucoup d'interactions</b>

	fonctionnel qui sera amélioré progressivement.	clients. - Permet d'obtenir un feedback rapide. - Réduction des risques d'échec.	nombreux prototypes sont nécessaires. - Risque que le prototype devienne le produit final sans une bonne structuration.	<b>utilisateurs</b> , nécessitant des ajustements fréquents (ex: applications web, logiciels UX/UI).
<b>Spirale</b>	Combinaison de l'incrémental et de la gestion des risques, basé sur des cycles de développement successifs.	- Gestion des risques dès le début. - Flexibilité pour intégrer des modifications. - Meilleure estimation des coûts et délais.	- Coût élevé en raison des analyses fréquentes. - Complexité de gestion. - Demande des compétences avancées en gestion de projet.	Projets <b>grands et complexes</b> , avec des incertitudes et des risques élevés (ex: développement de logiciels militaires, grandes applications d'entreprise).
<b>Incrémental</b>	Développement du projet par petits morceaux (incréments), chaque incrément apporte une nouvelle fonctionnalité.	- Livraison rapide de parties fonctionnelles du produit. - Facilité à intégrer des changements. - Réduction des risques par développement progressif.	- Peut être difficile d'intégrer des changements majeurs après plusieurs incréments. - Gestion plus complexe que le modèle en cascade.	Projets <b>évolutifs</b> , avec des exigences qui peuvent changer (ex: applications SaaS, logiciels d'entreprise évolutifs).