

# Table des matières

□ <b>G2 - Les modèles de gestion de projet</b> .....	3
□ <b>Panorama des modèles par famille</b> .....	3
□ <b>Modèles prédictifs (en cascade)</b> .....	4
□ <b>Modèles agiles</b> .....	5
□ <b>Modèles de gestion de formation spécifiques</b> .....	6
□ <b>Modèles standards en management de projet</b> .....	7
□ <b>Modèles hybrides</b> .....	8
□ <b>Comment choisir son modèle ?</b> .....	9



# G2 - Les modèles de gestion de projet

l'interdisciplinarité, l'apprentissage par projet et le travail de terrain.

## Informations rapides

- Page parente : [G2 - Concepts et méthodologies](#)
- Domaine : Management de projet / Ingénierie pédagogique
- Concept clé : Méthodologie de projet = cadre structurant l'action collective



G2 - Identifier et comparer les principales méthodologies de gestion de projet pour choisir celle adaptée à son contexte d'intervention

## Objectif de la fiche

Cette fiche présente un panorama des **principaux modèles de gestion de projet**, qu'ils soient issus du développement logiciel, du management général ou de l'ingénierie pédagogique. L'objectif est de disposer d'une vue d'ensemble pour **éclairer le choix méthodologique** en fonction du contexte, des contraintes et des besoins.



Le choix d'une méthodologie dépend de la taille du projet, des contraintes temporelles, de la flexibilité nécessaire et de la culture de l'organisation. Aucun modèle n'est universellement supérieur : chacun répond à des enjeux spécifiques.

## Panorama des modèles par famille

```
graph TD
  A["Modèles de gestion de projet (en cascade)"] --> B["Modèles prédictifs"]
  A --> C["Modèles agiles"]
  A --> D["Modèles formation spécifiques"]
  A --> E["Modèles standards management"]
  A --> F["Modèles hybrides"]
  B --> B1["Cycle en V"]
  B --> B2["Cycle en cascade"]
  C --> C1["Scrum - Kanban - XP - RAD"]
  D --> D1["ADDIE - SAM - Dick & Carey"]
  E --> E1["PMBOK - PRINCE2 - Cycle de vie"]
  F --> F1["Water-Scrum-Fall - Scrumban - DA"]
  style A fill:#e3f2fd,stroke:#2196f3,stroke-width:2px
  style B fill:#ffe0b2
  style C fill:#c8e6c9
  style D fill:#ffcdd2
  style E fill:#d1c4e9
  style F fill:#b2ebf2
```

## □ Modèles prédictifs (en cascade)

### □ Cycle en cascade

**Approche séquentielle linéaire** : Les phases se succèdent dans un ordre strict, chacune devant être complètement achevée avant de passer à la suivante.

#### Phases typiques :

1. Analyse des besoins
2. Conception
3. Réalisation / Développement
4. Tests
5. Déploiement
6. Maintenance

#### Avantages :

- Simplicité de compréhension et de gestion
- Jalons clairs et livrables bien définis
- Adapté aux projets stables avec exigences connues

#### Inconvénients :

- Rigidité face aux changements
- Retour utilisateur tardif
- Détection des erreurs en fin de cycle

### □ Cycle en V

**Version améliorée du cycle en cascade** avec validation à chaque étape. Chaque phase de développement est associée à une phase de test correspondante.

#### Structure :

- Analyse des besoins ↔ Tests d'acceptation
- Conception générale ↔ Tests d'intégration
- Conception détaillée ↔ Tests unitaires
- Codage / Réalisation

#### Avantages :

- Meilleure maîtrise de la qualité
- Validation progressive
- Traçabilité des exigences

#### Inconvénients :

- Toujours linéaire dans son déroulement
- Lourdeur documentaire potentielle

## □ Modèles agiles

### □ Méthode Scrum

**Organisation en sprints** avec des livraisons itératives et incrémentales.

#### Rôles clés :

- Product Owner (définit et priorise le besoin)
- Scrum Master (facilite le processus)
- Équipe de développement

#### Cérémonies :

- Sprint Planning (planification)
- Daily Scrum (point quotidien)
- Sprint Review (démonstration)
- Sprint Rétrospective (amélioration continue)

#### Artéfacts :

- Product Backlog (liste des fonctionnalités)
- Sprint Backlog (tâches du sprint)
- Incrément (livrable fonctionnel)

### □ Méthode Kanban

**Visualisation du flux de travail** avec gestion des tâches en continu, sans sprints prédéfinis.

#### Principes :

- Visualiser le travail (tableau Kanban)
- Limiter le travail en cours (WIP)
- Gérer le flux
- Rendre les processus explicites
- Améliorer en continu

#### Avantages :

- Flexibilité maximale
- Adaptation au rythme de l'équipe
- Idéal pour les activités de maintenance ou support

### ↪ Méthode eXtreme Programming (XP)

**Accent sur les bonnes pratiques et la qualité technique** du code.

#### Pratiques clés :

- Programmation en binôme (pair programming)
- Développement piloté par les tests (TDD)
- Intégration continue

- Refactoring permanent
- Client sur site
- Livraisons fréquentes

**Objectif :** Améliorer la qualité logicielle et la réactivité face aux changements.

□ [Méthode RAD \(Rapid Application Development\)](#)

**Développement rapide par prototypes** avec forte implication des utilisateurs.

**Phases :**

- Modélisation des données
- Modélisation des processus
- Génération d'applications
- Tests et livraison

**Caractéristiques :**

- Utilisation d'outils de prototypage
  - Itérations courtes
  - Feedback utilisateur immédiat
  - Adapté aux projets à forte composante interface utilisateur
- 

## □ **Modèles de gestion de formation spécifiques**

□ [Modèle ADDIE](#)

**Modèle de référence en ingénierie pédagogique :** Analyse, Design, Development, Implementation, Evaluation.

**Les 5 phases :**

- **Analyse** : Identification des besoins, du public, du contexte
- **Design** : Conception pédagogique, scénarisation
- **Development** : Production des ressources et supports
- **Implementation** : Mise en œuvre de la formation
- **Evaluation** : Mesure des acquis et de l'impact

**Caractéristique :** Approche séquentielle classique, stable et structurante.

□ [Modèle SAM \(Successive Approximation Model\)](#)

**Version itérative et agile d'ADDIE**, développée par Michael Allen.

**Principes :**

- Itérations rapides
- Prototypage précoce
- Collaboration étroite avec les experts
- Amélioration continue

**Phases :**

- Préparation (information gathering)
- Design itératif (prototypage)
- Développement itératif (amélioration)

**Avantage :** Plus flexible et adapté aux contextes où les besoins évoluent.

□ [Modèle Dick and Carey](#)

**Approche systémique et détaillée** de la conception pédagogique.

**Composantes :**

- Identification des buts instructionnels
- Analyse des objectifs
- Analyse des apprenants et du contexte
- Rédaction des objectifs de performance
- Développement des instruments d'évaluation
- Stratégie pédagogique
- Développement et sélection des ressources
- Évaluation formative et corrective
- Évaluation sommative

**Caractéristique :** Modèle très complet, adapté aux projets de formation complexes.

---

## □ **Modèles standards en management de projet**

□ [PMBOK \(Project Management Body of Knowledge\)](#)

**Référentiel de bonnes pratiques** du Project Management Institute (PMI).

**5 groupes de processus :**

- Démarrage
- Planification
- Exécution
- Surveillance et maîtrise
- Clôture

**10 domaines de connaissance :**

- Intégration, Périmètre, Délais, Coûts, Qualité
- Ressources, Communications, Risques, Achats, Parties prenantes

**Caractéristique :** Approche normative et exhaustive, adaptée aux grands projets.

□ [PRINCE2](#)

**Approche par processus orientée produit**, standard du gouvernement britannique.

## 7 principes :

- Justification continue
- Apprendre de l'expérience
- Rôles et responsabilités définis
- Management par phases
- Management par exception
- Focus sur les produits
- Adaptation au contexte

**Caractéristique** : Cadre méthodologique structuré, adaptable à différents contextes.

### [□ Modèle en cycle de vie](#)

**Phases structurées avec jalons** pour suivre l'avancement du projet.

### Phases classiques :

- Avant-projet (faisabilité, étude préalable)
- Lancement (cadrage, planification)
- Réalisation (exécution, suivi)
- Clôture (bilan, capitalisation)
- Exploitation (maintenance, évaluation)

**Caractéristique** : Vision temporelle globale du projet.

---

## □ Modèles hybrides

### [□ Water-Scrum-Fall](#)

**Combinaison du cascade et de l'agile** : planification en cascade, développement en Scrum, déploiement en cascade.

### Structure :

- Phases amont (analyse, spécifications) en cascade
- Phases de développement en Scrum (itérations)
- Phases aval (tests finaux, déploiement) en cascade

**Avantage** : Permet d'introduire de l'agilité dans des organisations traditionnellement en cascade.

**Risque** : Perte de cohérence si les interfaces entre phases sont mal gérées.

### [□ Scrumban](#)

**Mélange de Scrum et Kanban** pour bénéficier des deux approches.

### Caractéristiques :

- Garde la structure en sprints de Scrum (optionnelle)
- Utilise le tableau Kanban et la limitation du WIP

- Planning on demand (pas de sprint planning systématique)
- Rétrospectives régulières

**Avantage** : Flexibilité du flux avec la structure d'amélioration continue.

### □ Disciplined Agile

**Approche flexible selon le contexte** développée par Scott Ambler.

#### Principes :

- Choisir son niveau de cérémonie selon le contexte
- Adapter les pratiques plutôt que les appliquer mécaniquement
- Se concentrer sur la valeur
- Optimiser le flux

**Caractéristique** : Méta-méthodologie qui emprunte à Scrum, Kanban, XP, etc., selon les besoins.

## □ Comment choisir son modèle ?

Critère	Modèles prédictifs	Modèles agiles	Modèles hybrides
<b>Stabilité des besoins</b>	Élevée	Faible / Évolutive	Variable
<b>Taille du projet</b>	Grande équipe	Petite/moyenne équipe	Toute taille
<b>Criticité</b>	Élevée (sécurité, réglementaire)	Faible à modérée	Variable
<b>Implication client</b>	Ponctuelle (jalons)	Continue	Selon phases
<b>Culture organisationnelle</b>	Traditionnelle	Innovante	En transition
<b>Délais</b>	Connus à l'avance	Itérations courtes	Mixte



**Règle d'or** : Aucun modèle n'est parfait en soi. L'important est de **comprendre les principes sous-jacents** pour les adapter à son contexte, plutôt que d'appliquer une méthode de manière dogmatique.

### □ Synthèse rapide

#### Ce que j'ai appris

- Il existe une grande diversité de modèles, chacun avec ses forces et ses limites.
- Les modèles prédictifs rassurent par leur cadre mais manquent de flexibilité.
- Les modèles agiles favorisent l'adaptation mais exigent une maturité d'équipe.
- Les modèles hybrides tentent de concilier les avantages des deux mondes.
- Le choix dépend du contexte, pas d'une vérité absolue.

### □ Pistes de développement

#### Ce que je veux améliorer

- Savoir diagnostiquer le contexte pour recommander un modèle adapté.
- Maîtriser les fondamentaux de Scrum et Kanban, très répandus.
- Comprendre comment adapter ADDIE avec des principes agiles (SAM).
- Éviter le “zombie methodism” : appliquer une méthode sans comprendre pourquoi.

## □ [Réflexion personnelle](#)

### Mon cheminement avec ces modèles

La diversité des modèles m'a longtemps semblé déroutante : pourquoi tant de méthodes pour “simplement” gérer un projet ? Je comprends maintenant que cette diversité reflète la variété des contextes, des cultures et des enjeux. Un projet de formation n'a pas les mêmes contraintes qu'un projet de développement logiciel critique.

Ce qui me paraît essentiel, c'est de distinguer **l'esprit de la lettre**. L'esprit de l'agilité (collaboration, adaptation, valeur) peut inspirer même un projet en apparence très cadré. Inversement, certains projets agiles gagneraient à emprunter aux modèles prédictifs leur rigueur sur les livrables.

En tant que future professionnelle de la formation, je retiens particulièrement l'intérêt des modèles hybrides comme SAM, qui conservent la rigueur d'ADDIE tout en introduisant de l'itératif. C'est probablement dans cet équilibre que je trouverai ma posture : suffisamment de cadre pour sécuriser, suffisamment de souplesse pour m'adapter.

---

## □ **Bibliographie**

- Project Management Institute. (2021). \*Guide du corpus des connaissances en management de projet (PMBOK Guide)\* (7e éd.). PMI.
- Allen, M. (2012). \*Leaving ADDIE for SAM : An Agile Model for Developing the Best Learning Experiences\*. ASTD Press.
- Dick, W., Carey, L., & Carey, J. O. (2014). \*The Systematic Design of Instruction\* (8e éd.). Pearson.
- Beck, K. (2000). \*Extreme Programming Explained : Embrace Change\*. Addison-Wesley.
- Schwaber, K., & Sutherland, J. (2020). \*The Scrum Guide\*.

---

## □ **Navigation**

- ← Retour à [G2 - Concepts et méthodologies](#)

---

[g2](#), [projet](#), [méthodologie](#), [agile](#), [cycle en v](#), [scrum](#), [kanban](#), [addie](#), [pmbok](#), [prince2](#)

Page mise à jour le {{date | Auteur : Eugénie Decré | Version : 1.0}}

From:

<https://wiki.eugeniedecre.com/> - **Formation en Conscience**

Permanent link:

[https://wiki.eugeniedecre.com/doku.php?id=carnet:g2\\_competences:g2\\_projet](https://wiki.eugeniedecre.com/doku.php?id=carnet:g2_competences:g2_projet)

Last update: **2026/03/03 10:14**

