



Activité Préapprentissage

# Architecture & Conception de logiciels

TheDevelopers



# Plan

- 1 Les critères de qualité de la conception architecturale
- 2 Atelier patrons de conception



# Les membres d'équipe

TheDevelopers



**Rania Wachene**



**Meriam Wachene**



**Houyem Haj  
Mansour**



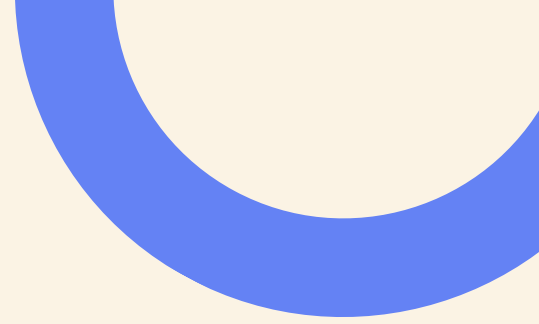
**Ahmed Boutaba**





**Wassim Manai**



# **Les critères de qualité de la conception architecturale**

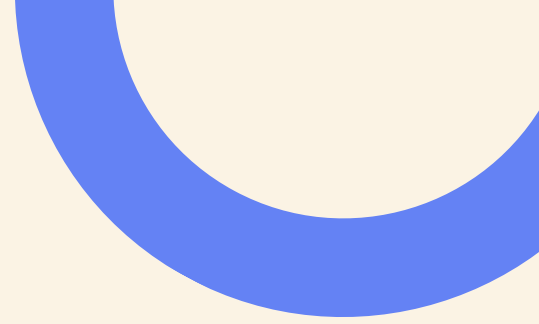


Les critères de qualité de la conception architecturale incluent généralement le couplage et la cohésion, qui sont deux concepts clés en matière de conception de logiciels.





# 1- Couplage :

- Niveau d'interaction entre composantes
  - Par données (paramètres ou classes)
  - Par contrôle (indique comment s'exécuter)
    - Externe (fichier, lien de communication)
  - Un couplage faible signifie que les modules sont relativement indépendants les uns des autres
  - couplage élevé signifie qu'ils sont fortement interdépendants.



En général, un couplage faible est considéré comme un critère de qualité, car il rend le système plus facile à maintenir, à modifier et à tester.

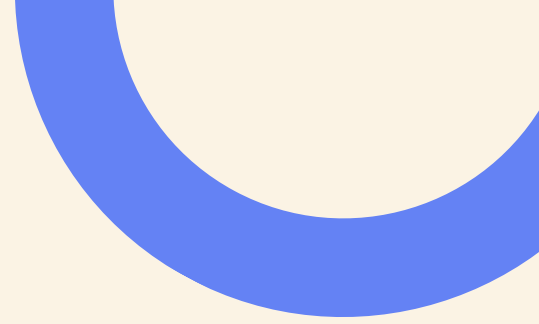






## 2- Cohésion :

- Lien entre les responsabilités d'un module
  - Logique
  - Temporelle et procédurale
  - Communicationnelle
  - Séquentielle et fonctionnelle
- cohésion forte signifie que les fonctionnalités sont étroitement liées et ont un objectif commun
- cohésion faible signifie qu'elles sont plus disjointes et moins liées.





En général, Une cohésion forte est également considérée comme un critère de qualité, car elle rend les modules plus faciles à comprendre, à maintenir et à tester.





# **Atelier patrons de conception**



# Patron de conception : prototype

**La catégorie du patron de conception :** Création

**Objectif de la catégorie:** Cette catégorie de conception vise à fournir des moyens flexibles pour créer des objets et des classes. Les patrons de cette catégorie offrent des méthodes pour instancier des objets tout en minimisant la duplication de code et en maximisant la réutilisabilité du code.

**Patrons de conception :** Singleton et Fabrique



# Patron de conception : prototype

**Problématique :** le patron de conception Prototype permet de créer des objets de manière efficace en évitant la duplication de code, en maximisant la réutilisabilité et en facilitant la maintenance du code.

**Solution :** permet de créer des objets de manière flexible et réutilisable en utilisant un modèle de base pour créer de nouvelles instances en les clonant. Cela permet de minimiser la duplication de code et de maximiser la réutilisation de code existant.



# Patron de conception : Façade

**La catégorie du patron de conception :** Conception structurelle

**Objectif de la catégorie:** fournir des solutions pour améliorer la structure interne d'un système et faciliter les interactions entre ses différentes parties.

**Patrons de conception :** Le patron de conception Adapter et Le patron de conception Composite



# Patron de conception : Façade

**Problématique:** résout la problématique de la complexité et de la difficulté de compréhension d'un système en fournissant une interface unifiée et simplifiée pour accéder à ses différentes parties.

**Solution:** simplifier l'interaction avec un système complexe en fournissant une interface unifiée et simplifiée pour accéder à ses différentes parties.



# Patron de conception : Observer

**La catégorie du patron de conception :** Conception comportementale.

**Objectif de la catégorie:** modéliser et de faciliter les interactions dynamiques entre les objets en définissant leurs comportements et leurs collaborations à travers des patrons de conception.

**Patrons de conception :** Patron de conception Stratégie et Patron de conception Command





# Patron de conception : Observer

**Problématique :** résout le problème de la communication flexible et décentralisée entre des objets qui doivent être notifiés lorsqu'un autre objet change d'état, sans imposer des dépendances rigides et directes entre eux.

**Solution :** permet de mettre en place une relation de type "un-à-plusieurs" entre les objets d'un système, où un objet (appelé "sujet") est capable de notifier automatiquement plusieurs autres objets (appelés "observateurs") lorsqu'un événement se produit ou lorsqu'il change d'état.

# Personnages de l'Histoire



Protagoniste

---

Antagoniste

---

Personnages Tertiaires

---



# Personnages de l'Histoire

Protagoniste



Antagoniste



Personnages Tertiaires



# Cadre de l'Histoire



Cadre Temporel

---




Lieu

---



# Séquence d'Événements

Exposition :

A large, empty white rectangular box with rounded corners, intended for writing the 'Exposition' part of the event sequence.

Conflit

A large, empty white rectangular box with rounded corners, intended for writing the 'Conflit' part of the event sequence.



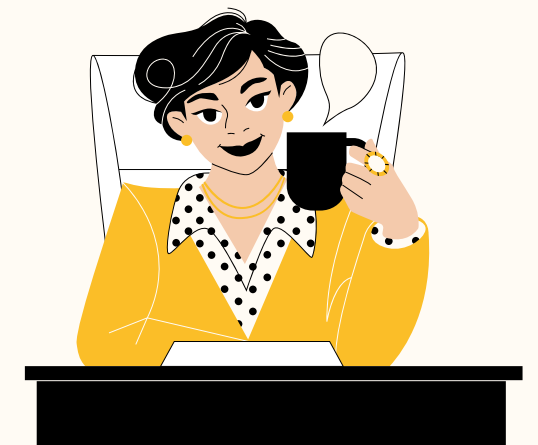
# Merci !

Avez-vous des  
questions de suivi pour moi ?



# Page de Ressources pour les Étudiants

Utilisez ces icônes et illustrations  
dans votre présentation Canva.  
Bonne création !





# Page de Ressources pour les Étudiants

Trouvez la magie et le plaisir de présenter avec les présentations Canva. Appuyez sur les touches suivantes en mode Présent !

**B pour flou**

**C pour les confettis**

**D pour un roulement de tambour**

**O pour les bulles**

**Q pour calme**

**X pour fermer**

**N'importe quel chiffre de 0 à 9 pour une minuterie**