

Génie Logiciel :

Logiciel : un programme info pour résoudre un problème

Le génie logiciel : est l'art et la manière de bien produire de bons logiciels au meilleur rapport qualité/prix

→ CQFD: cout, qualité , fonctionnalité , delais

Les auteurs d'un projet :

→ MOA: Le maître d'ouvrage

→ MOE: Le maître d'œuvre

→ Les intervenants

⇒ **les étapes a suivre avant le projet :**

1. étude de faisabilité: est ce qu'on est capable de réaliser ce projet?

Faisabilité technique:

Risques de développement / Disponibilité des ressources / Technologie nécessaire

Faisabilité economique :

Analyse du rapport Coût/Bénéfice :

Coût du système

Bénéfices mesurables (en DH)

Bénéfices non mesurables

- meilleure conception
- meilleures décisions marketing
- Satisfaction accrue du client

L'analyse Coût/Bénéfice est souvent le moyen d'obtenir le feu vert de la direction

Estimation:

Estimer la taille du produit à développer.

Estimer la charge en mois hommes ou en jours hommes

Construire le calendrier du planning

Estimer le coût du projet en monnaie locale

2. analyse de besoins

un besoin ? : ce qui est exprimé ou non exprimé par l'utilisateur et qui lui permet de réaliser un certain nombre d'actions ou tâches.

L'analyse du besoin consiste à traduire ce qu'exprime l'utilisateur dans un document qui permettra la réalisation d'un produit ou service qui satisfera le besoin.

3. cahier des charges:

definition: est un document par lequel on exprime son besoin pour le projet contenant une Description globale des fonctions d'un nouveau produit ou des extensions à un produit existant

IL SE COMPOSE DE 4 parties:

- **rapport de cadrage** : tactiques que l'entreprise va suivre
- **planning détaillé** : gant
- **plan assurance qualité** : mesure de qualité pour juger la qualité
- **architecture technique**: rapport des composants , fonctionnalités , frameworks ... utilisés

Méthodes classiques :

la réalisation d'un projet :

- Étape 1 : définition
- Étape 2 : développement
- Étape 3 : support

-> Les Activités lors de ces 3 étapes

- spécification des besoins : CDC
- Analyse (analyst) : (diagrammes d'analyse/classes)
- Conception (architect) : décomposer le system (diagrammes d'interaction, séquences, composant, déploiement)
- codage : transforme les solutions
- tests : tester le logiciel
- Maintenance : après la livraison (correctif, adaptative, préventif)

Processus de développement

Un processus de développement logiciel est un ensemble d'activités conduisant à la production d'un logiciel. aussi appelés cycle de vie d'un logiciel.

Objectif :

- Maîtriser les gros projets
- Découper le projet et affecter correctement les tâches
- Anticiper et gérer les risques

les principaux modeles:

Approche classique :

Modele	principe	Avantage	Inconvénients
En cascade/ lineaire	Considérer le développement logiciel comme une succession d'étapes réalisées de façon strictement séquentielle.	Facile à utiliser et à comprendre Idéal pour la gestion et le suivi de projets	-Sensibilité aux nouveaux besoins : refaire tout -Une phase ne peut démarrer que si l'étape précédente est finie -Le produit n'est visible qu'à la fin -Les risques se décalent vers la fin -on code d'abord et on modifie ensuite -Inadapté aux développements en équipe ou de grande taille
En V	Variante du modèle en cascade qui fait l'accent sur la vérification et la validation Le test du produit se fait en parallèle par rapport aux autres activités	-Met l'accent sur les tests et la validation et par conséquent, ça accroît la qualité du logiciel -Chaque livrable doit être testable -Facile à planifier dans une gestion de projets -Facile à utiliser	Ne gère pas explicitement les changements des spécifications
En Y	Son intérêt est de séparer les préoccupations concernant les aspects fonctionnels liés au domaine métier et les aspects techniques liés aux solutions technologique à employer pour la mise en œuvre		

Approche Agile :

définition:

L'approche agile est une méthodologie de gestion de projet qui met l'accent sur la collaboration, l'adaptabilité et la livraison rapide et incrémentale de produits ou de services. Cette approche est souvent utilisée dans le développement de logiciels, mais peut également être appliquée à d'autres domaines.

Une méthode agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif, avec juste ce qu'il faut de formalisme.

- incrémental : décomposer le projet en itérations
- itératif : chaque incrément dans l'itération
- Adaptatif : accepter les changements

Les 12 principes :

- Satisfaire le besoin du client
- Accepter les changements
- Livraison régulière du logiciel
- Collaboration étroite entre client et dev
- Confiance
- Communication directe
- Partager l'état d'avancement
- Rester motivé (même rythme)
- Qualité technique (Refactoring)
- Simplicité
- Auto organisation et la collaboration
- Amélioration continue

Agile = itératif + incrémental + adaptatif

SCRUM

Différence entre cascade et scrum :

- cascade est un processus de Dev

- scrum est une méthode de gestion de projet doit être complémentaire avec des Tech d'ingénieur

Scrum n'est pas un processus ou une méthode mais une voie à suivre.

Définition:

Scrum est un cadre de développement de logiciels itératif et incrémental utilisé pour gérer et réaliser des projets complexes. Il se concentre sur la collaboration d'équipe, la communication, la flexibilité et la livraison continue de fonctionnalités à valeur ajoutée pour le client.

les valeurs :

- Commitement : engagement
- Focus : concentration
- Openness: l'ouverture
- Respect: respect des dates
- Courage

Répartition des rôles :

-Le Product Owner : Il détermine les caractéristiques du produit, fixe la date de lancement et définit le Product Back log.

-Le ScrumMaster est le chef de projet Scrum. Il organise les réunions et assure une collaboration étroite avec le Product Owner .

-Team members ce sont les personnes chargées de la réalisation du sprint

Concepts de SCRUM:

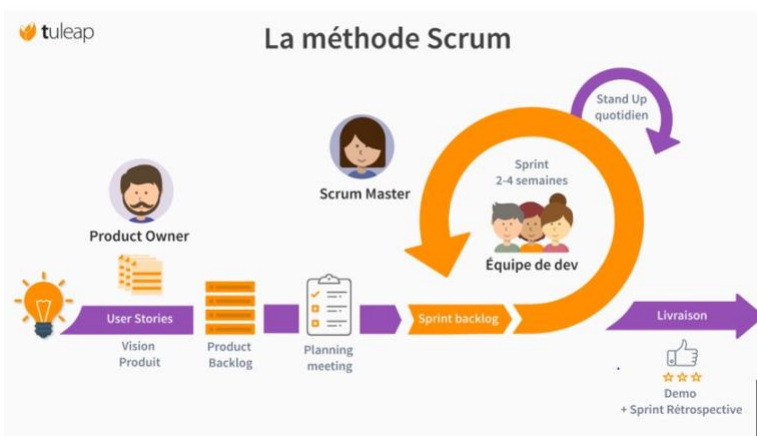
-Le Product backlog : représente le référentiel des exigences initiales est dressé et hiérarchisé avec le client.

-Un Sprint : est une itération de 15 à 30 jours.

-Le Sprint backlog : est un plan d'itération . Il contient la liste de tâches (user stories) que l'équipe s'engage à livrer à la fin du sprint (un sous ensemble du product backlog).

- Le Sprint Burn-down : combien d'effort et le comparer a la dépense idéale
- Planning Poker : Il s'agit ici du chiffrage des user stories en points d'efforts (ou niveau de complexité).
- découpage des tâches : attribuer les tâches (modele canban)
- sprint planning meeting : Réunion de planification d'un Sprint
- Le Scrum daily meeting : est une réunion quotidienne de 15 minutes.
- Le Sprint review meeting : présenter les résultats tout les participants
- Le Sprint retrospective meeting : est une réunion pour evaluer les efforts

Organisation :



XP (extreme programming): est une approche tout à fait réfléchie et disciplinée du développement logiciel.

cycle de vie xp :

- Phase d'exploration : déterminer ce qui doit être implémenté au cours de l'itération.
- Phase de planification : l'équipe transforme les scénarios en à réaliser.
- Phase de mise en production : chaque développeur procède à la réalisation des tâches avec un binôme.

- Phase de maintenance : tests fonctionnels sont exécutés et le produit est livré.

valeurs:

communication

simplicité

feed-back

courage

pratiques:

l'équipe:

Programmation en binôme (pair programming) :

Responsabilité collective du code

Règles de codage (coding standards)

Métaphore (metaphor)

Intégration continue (continuous integration)

gestion de projet

Livraisons fréquentes (frequent releases)

Planification itérative (planning game)

Client sur site (on-site customer, whole team)

Rythme durable (sustainable pace)

Rôles : programmeur, testeur, tracker, client, manager, coach

DESIGN PATTERNS

définition :

Les design patterns, sont des solutions réutilisables générales aux problèmes couramment rencontrés en conception de logiciels. (<problème/solution>)

Types :

- Créateurs (Création d'objets) : Singleton , Factory
- Structure (Structure des relations entre objets) : Décorateur
- Comportement (Comportement des objets) : Observateur

Définition du couplage :

⇒ Mesure de dépendance entre les classes

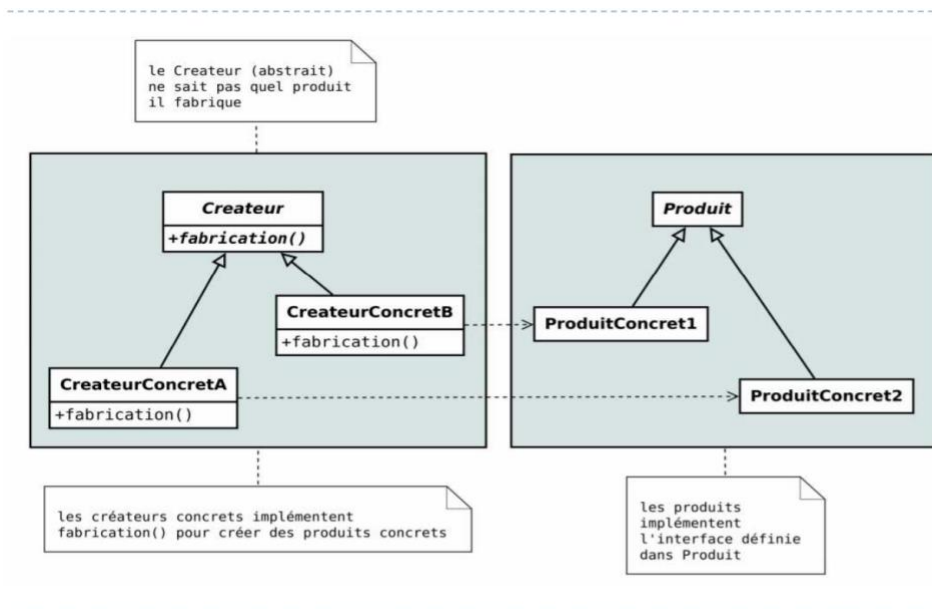
Le couplage est un concept qui décrit la mesure de l'interdépendance entre deux modules ou composants logiciels

types :

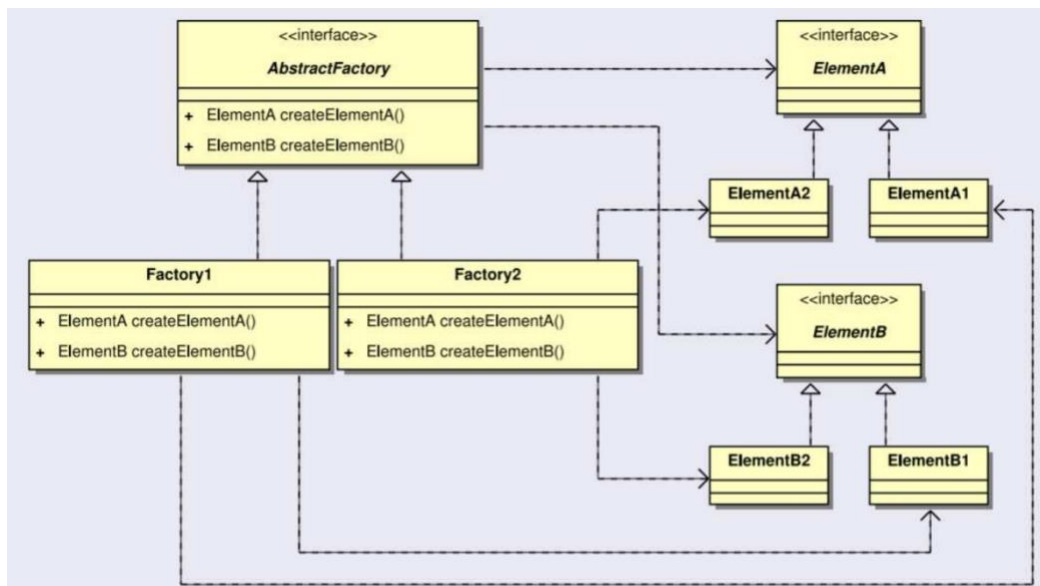
- ⇒ Le faible couplage a pour objectif de faciliter la maintenance en minimisant les dépendances entre éléments.
- ⇒ un fort couplage peut rendre le système plus difficile à maintenir, à modifier et à adapter à de nouveaux besoins

GOF : Gang Of Four (23 pattern)

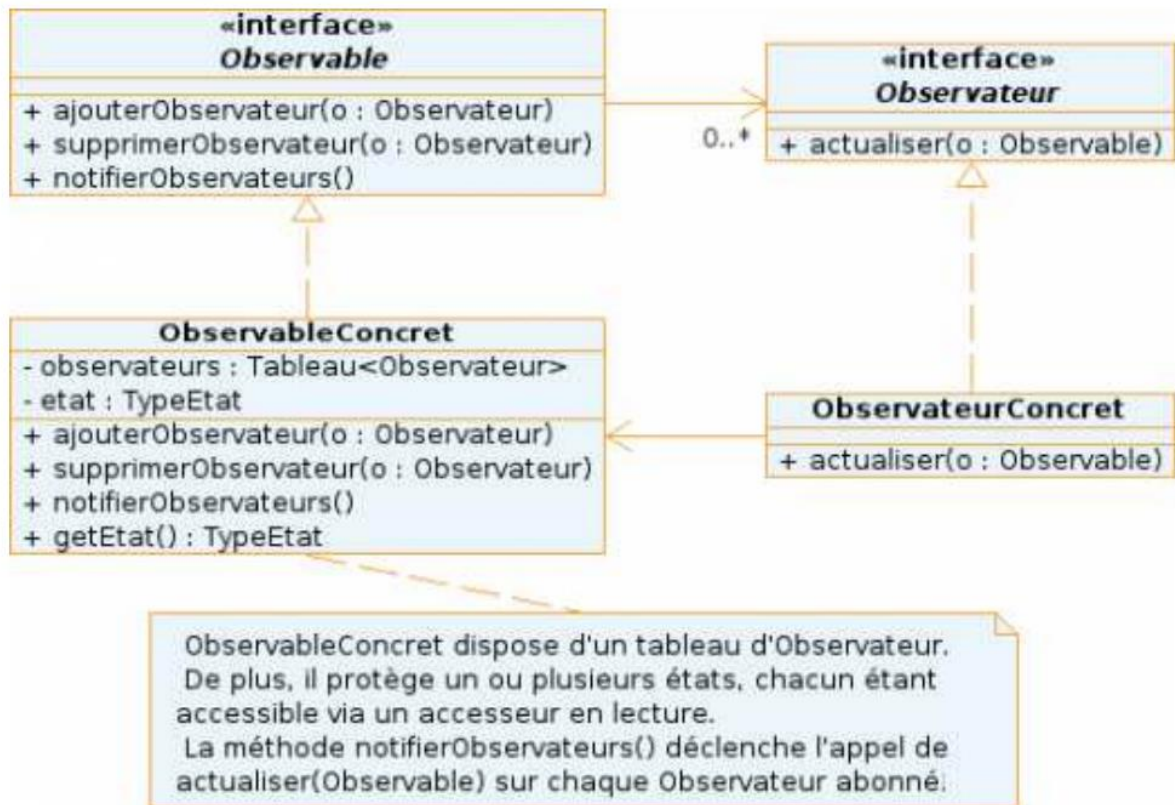
- ⇒ **Singleton** : une seule instance d'une classe pour toute la durée de l'app
- « Simple Factory : regrouper l'instanciation de tous les objets dans une seule classe »
- ⇒ **Factory** : une classe responsable de fabriquer les objets d'une autre classe



⇒ **Abstract Factory** : création d'objets regroupés en familles sans devoir connaître les classes concrètes destinées à la création de ces objets



⇒ **Observateur** : définit une relation entre objets de façon que si un objet change d'état tous ceux qui en dépendent en soient informés et mis à jour auto



⇒ Decorateur

L'Abstract Factory Pattern est utilisé pour retourner un groupe de Classes.