

Chapitre 1

Modélisation d'un projet web

Module : Préparation d'un projet Web

Formateur : Mahdi KELLOUCH

TABLES DES MATIÈRES

1. Projet web
2. Processus de développement
3. Modélisation
4. UML
5. Atelier 01 - Travail de recherche

1. Projet WEB

Modélisation d'un projet web

Définitions

- **Logiciel** : Un logiciel ou une application est un ensemble de programmes, qui permet à un ordinateur ou à un système informatique d'assurer une tâche ou une fonction en particulier (exemple : logiciel de gestion de la relation client, logiciel de production, logiciel de comptabilité, logiciel de gestion des prêts).
- **Application Web** : est un logiciel hébergé sur un serveur, à laquelle on accède par un navigateur et une connexion internet.
- **Site Web** : est un ensemble de contenus, que vous pouvez parcourir, consulter et lire. Il ne propose pas, en lui-même, d'interactions à l'utilisateur : vous pouvez simplement naviguer entre les différents contenus.
- **Projet Web** : un projet informatique dont les livrables (résultats concrets attendus) ont pour destination finale, le réseau Internet.

PROJET WEB

Un projet Web doit :

- Répondre à un besoin fonctionnel précis.
- Mobiliser les ressources nécessaires (humaines, techniques, technologiques, matérielles).
- S'inscrire dans la limite de l'enveloppe budgétaire allouée.
- Respecter un calendrier précis.

LES OBJECTIFS D'UN PROJET WEB

- Avant de démarrer un projet web, il est impératif pour l'entreprise de définir clairement ce que l'on attend concrètement de son projet web.
- Voici quelques objectifs fondamentaux :
 - Améliorer votre notoriété.
 - Vendre vos produits et services en ligne
 - Acquérir de nouveaux clients.
 - Informer au sujet de votre entreprise
 - Fidéliser votre clientèle

2. Processus de développement

Modélisation d'un projet web

Les processus de développement

La qualité du **processus de fabrication** est
garante de la **qualité du produit**

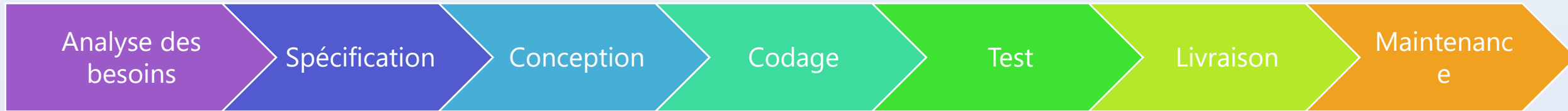
Processus de développement

Définition : Ensemble d'activités successives, organisées en vue de la production d'un logiciel.

En pratique :

- Pas de processus idéal
- Choix du processus en fonction des contraintes (taille des équipes, temps, qualité...)
- **Pour obtenir un logiciel de qualité**, il faut en maîtriser le processus d'élaboration
 - La vie d'un logiciel est composée de différentes étapes.
 - La succession de ces étapes forme le cycle de vie du logiciel
 - Il faut contrôler la succession de ces différentes étapes

Les étapes de cycle de vie

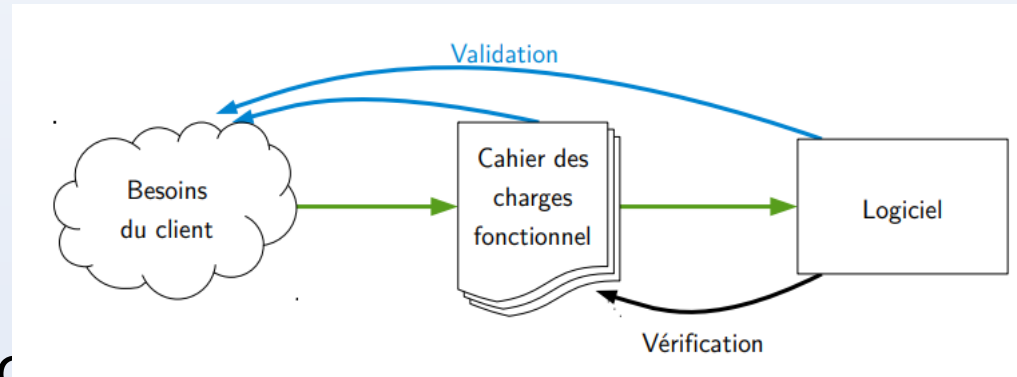


Les étapes de cycle de vie

1. **Analyse des besoins** : consiste à recueillir et à documenter de manière détaillée les attentes, les exigences et les fonctionnalités souhaitées par les utilisateurs d'un projet.
2. **Spécification** : Déterminer les fonctionnalités que doit posséder le logiciel.
3. **Conception** : Déterminer la façon dont le logiciel fournit les différentes fonctionnalités recherchées.
4. **Implémentation** : Implantation de la solution conçue
 1. • Choix de l'environnement de développement, du/des langage(s) de programmation, de normes de développement...

Les étapes de cycle de vie

5. Test : assurer que les besoins du client sont satisfaits (au niveau de la spécification, du produit fini...)

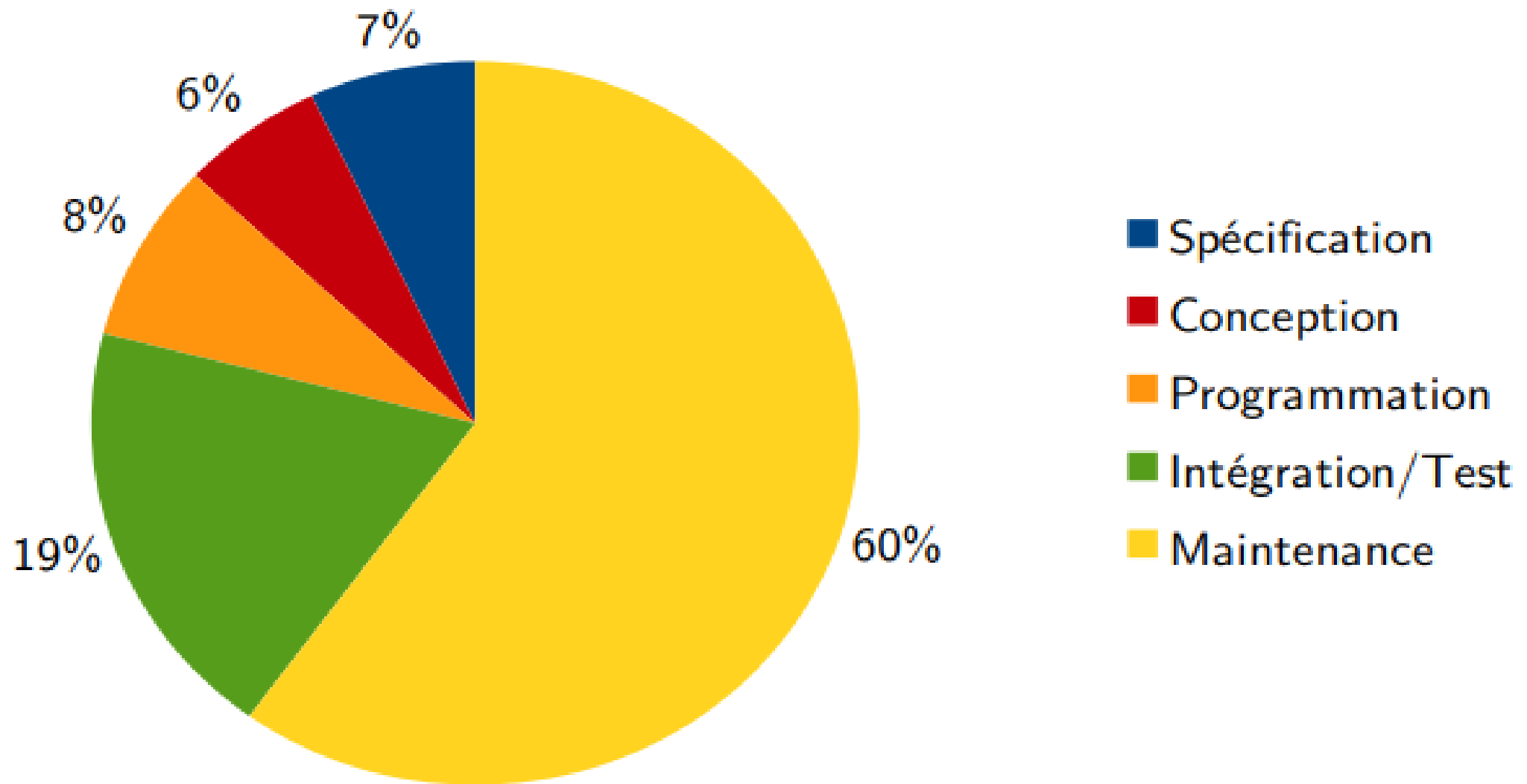


6. Livraison : Fournir au client une solution logicielle qui fonctionne correctement :

1. Installation : rendre le logiciel opérationnel sur le site du client
2. Formation : enseigner aux utilisateurs à se servir du logiciel
3. Assistance : répondre aux questions des utilisateurs

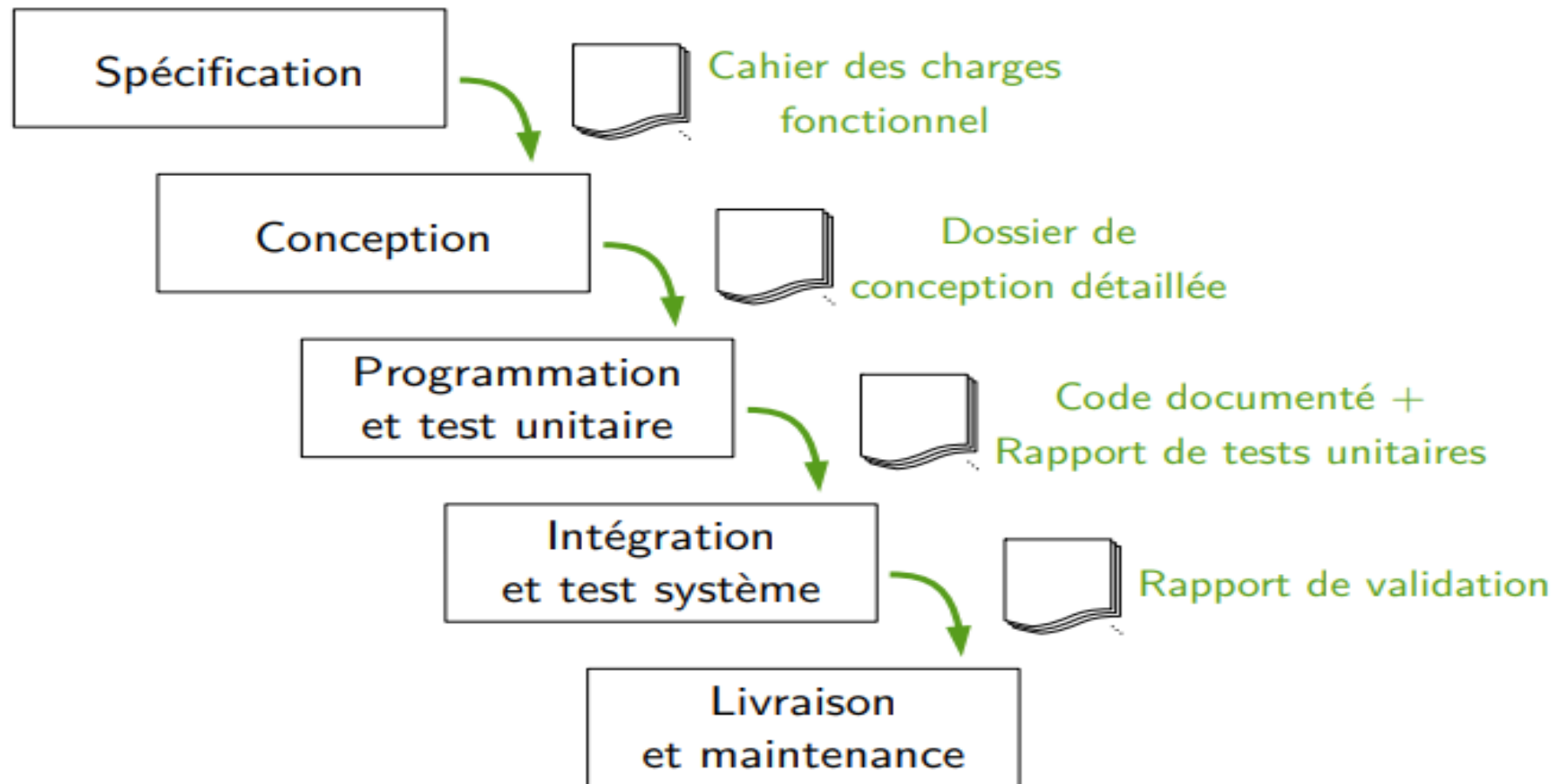
7. Maintenance : Mettre à jour et améliorer le logiciel pour assurer sa pérennité.

Répartition de l'effort

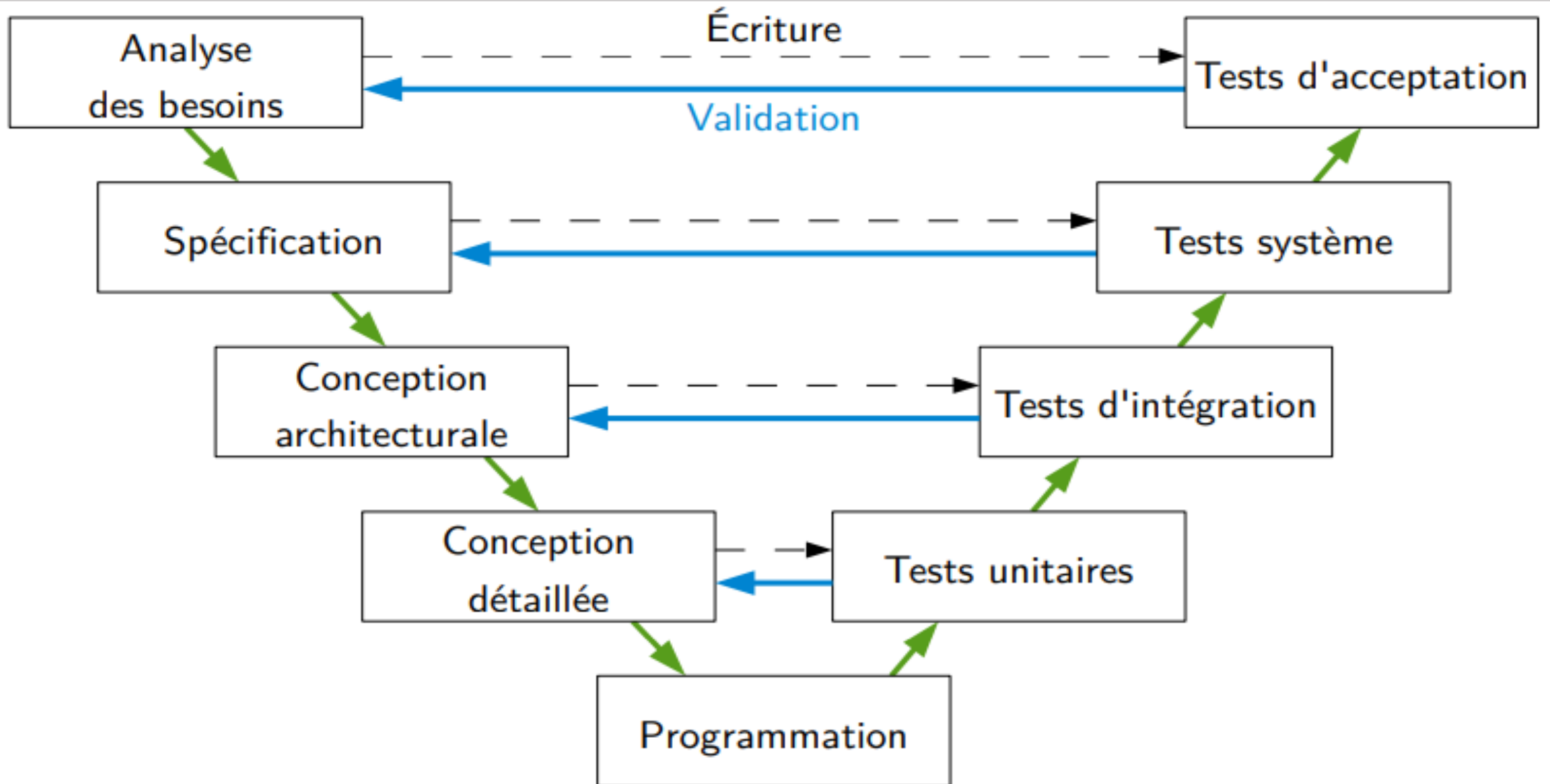


Modèle en cascade

- Chaque étape doit être terminée avant que ne commence la suivante À chaque étape, production d'un document base de l'étape suivante

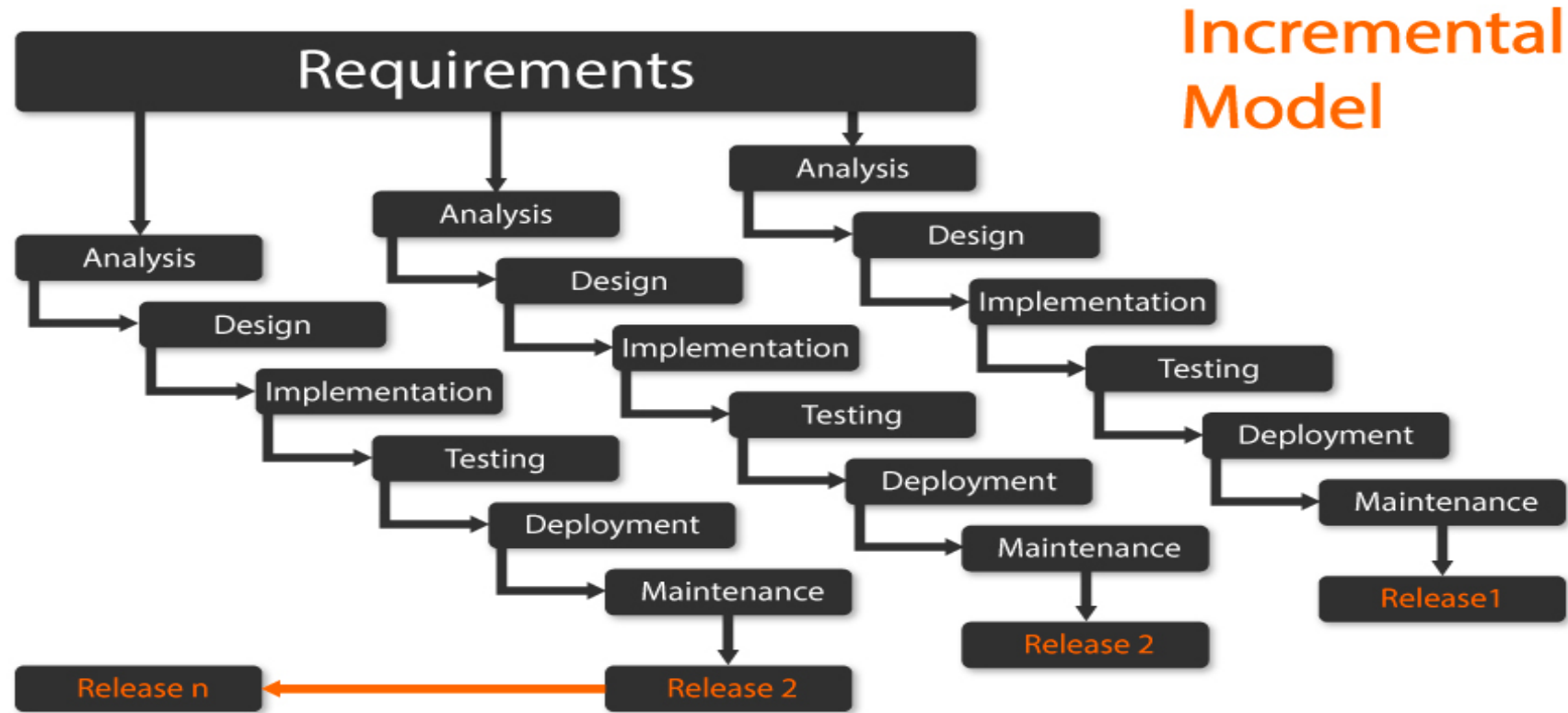


Modèle en V



Modèle incrémental

- Plutôt que de livrer un système en une fois, le développement et la livraison sont découpés en incréments, chaque incrément permettant de livrer une partie de la fonctionnalité



Modèle de prototypage

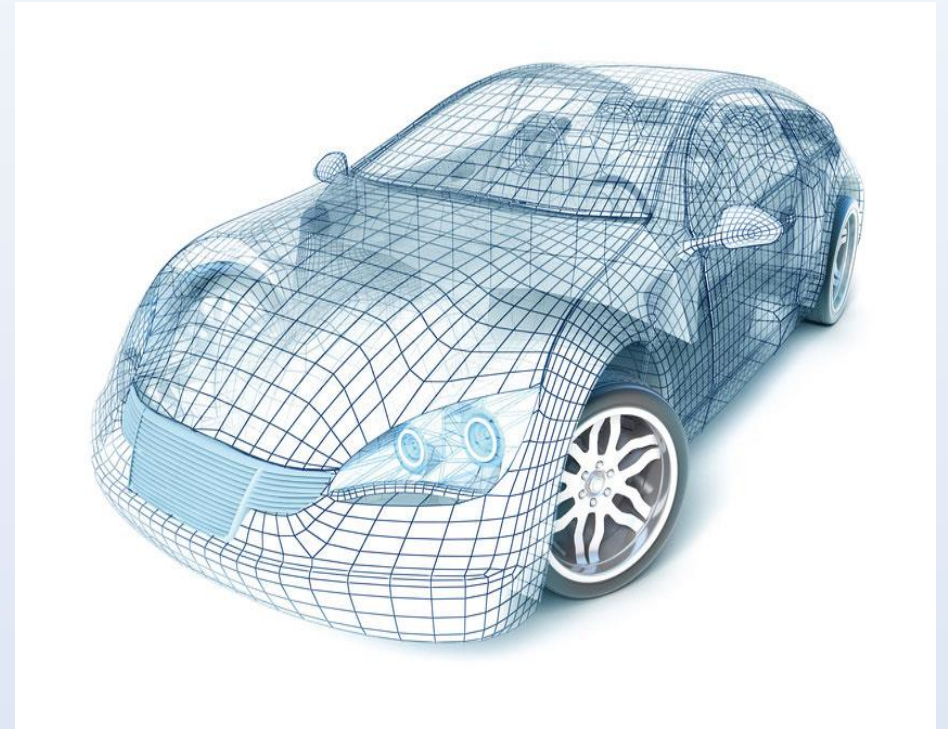
- Un prototype est une version initiale/intermédiaire d'un système, utilisée pour démontrer des concepts et faire des essais de choix de conception.

4. Modélisation

Modélisation d'un projet web

Introduction

- La modélisation consiste à représenter d'une façon simplifiée le monde réel.
- La modélisation est la représentation d'un système par un autre, plus facile à appréhender.
- Un modèle est une représentation graphique simplifiée et abstraite (exclut les détails) d'une partie ou de la totalité d'un système réel.



Pourquoi modélisation ?

- Il permet :
 - Mieux comprendre :
 - Le système à développer.
 - Comprendre et de décrire les besoins.
 - Concevoir et construire les solutions possibles.
 - Faciliter la communication : entre les acteurs (concepteurs, développeurs, clients...) impliqués dans le développement du projet grâce aux illustrations graphiques et symboliques du projet futur.
 - Réduire la complexité : Elimination des détails qui n'influencent pas la compréhension du système.

3. UML

Modélisation d'un projet web

C'est quoi ?

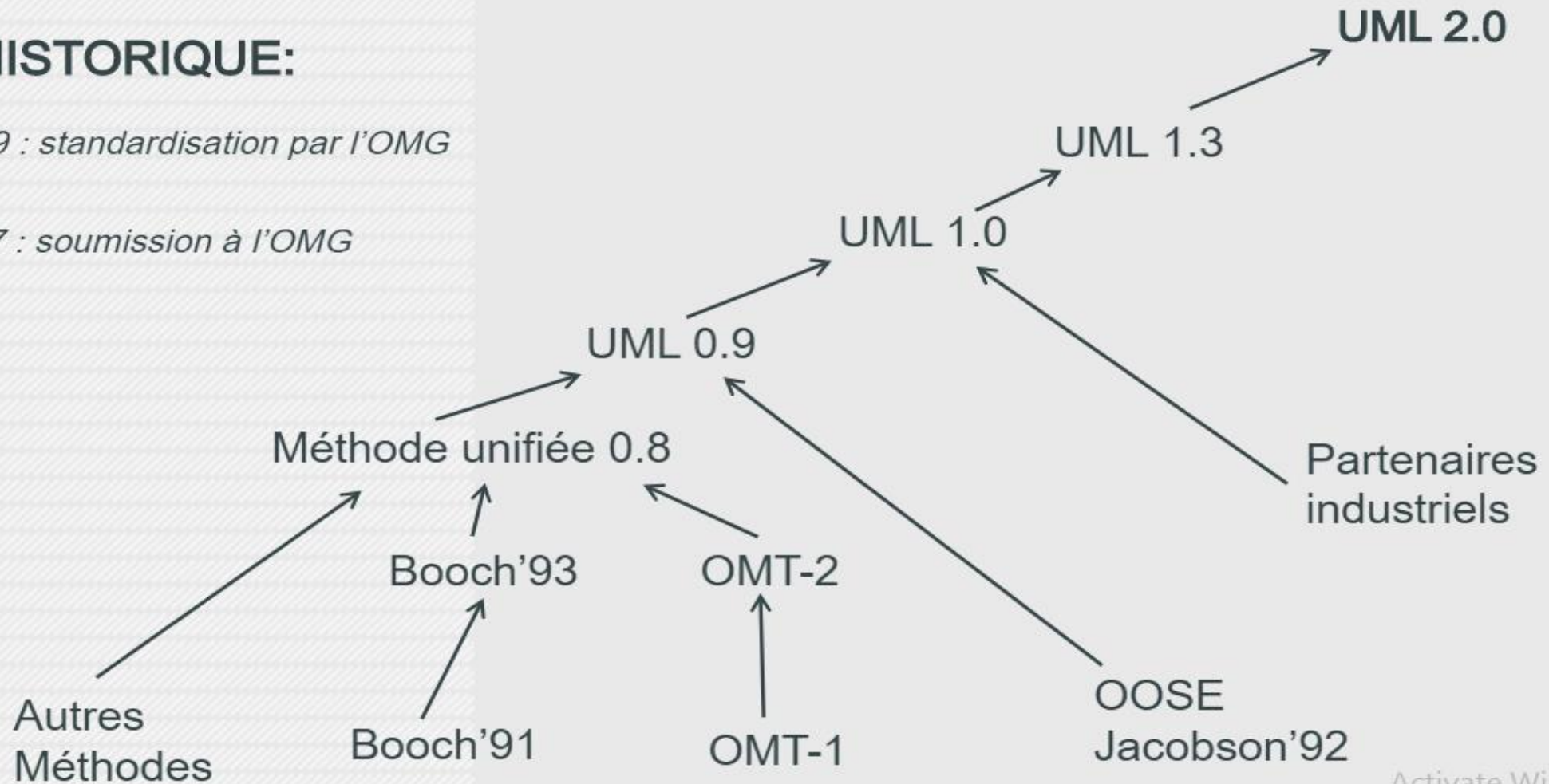
- UML, c'est l'acronyme anglais pour « **Unified Modeling Language** ». On le traduit par « Langage de modélisation unifié ».
- La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter.
- UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc.
- **Réaliser ces diagrammes revient donc à modéliser les besoins du logiciel à développer.**

SENSIBILISATION A LA MODELISATION

HISTORIQUE:

1999 : standardisation par l'OMG

1997 : soumission à l'OMG



Les modèles d'UML

- Modèles UML sont manipulés au moyen de vues graphiques : 9 diagrammes essentiels:
 - **Diagrammes des cas d'utilisation.**
 - **Diagrammes de séquence.**
 - **Diagrammes de classes.**
 - **Diagrammes de collaboration**
 - **Diagrammes d'objets.**
 - **Diagrammes d'états-transitions**
 - **Diagrammes d'activités**
 - **Diagrammes de composants**
 - **Diagramme de déploiement**