

L'École Supérieure d'Ingénierie Appliquée
et Innovation Privée (ESIAI)

Gestion de projets

Pr. Sanae MAZOUZ
2025-2026



Plan

- Introduction aux **concepts de base**
- **Phases** d'un projet: avant-vente, définition, conception, réalisation, recette , déploiement
- **Cycles de vie** des projets informatiques : cascade / v , itératif
- **Gestion des exigences** : product backlog
- **Estimation** de charge et du coût d'un projet
- **Gestion des risques**
- **Planification** du projet : planning prévisionnel
- **Proposition de la réponse technique et financière** pour un appel d'offre / cahier de charges
- **Méthodologies agiles (SCRUM,SAFe, KANBAN..)**
- **Collaboration** en équipe : **Git**
- **Gestion de la qualité**
- **Approche DevOps**
- Outils: Jira, Gantt project , GitHub

2) Phases de réalisation d'un projet en Informatique “Cas d'un projet en ingénierie logicielle”

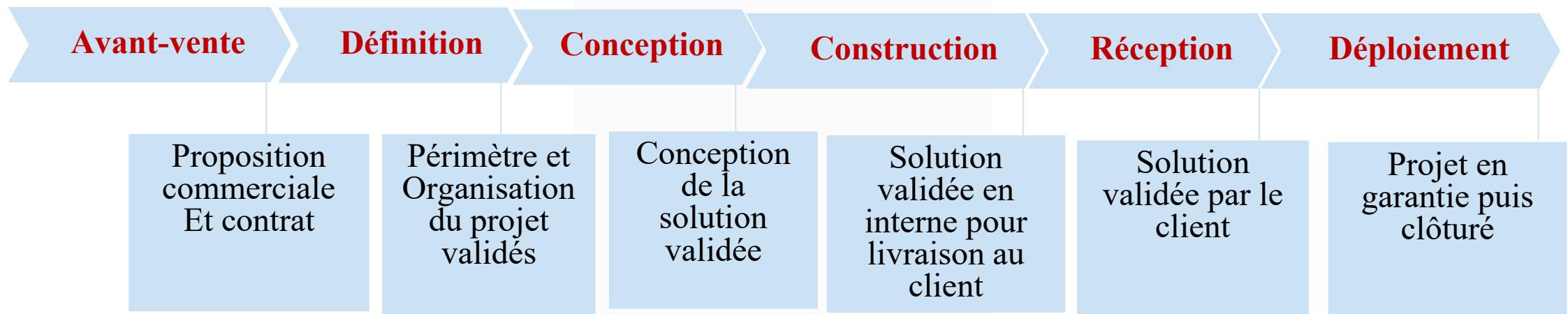


- Avant-vente
- Définition
- Conception
- Construction
- Réception
- Déploiement

Les phases d'un projet informatique

Selon le guide “Software extension to the PMBOK guide, fifth edition-Project Management Institute (2013)” et selon le modèle CMMI,

On peut lister les phases de réalisation d'un projet en génie logiciel comme suit :



Les phases d'un projet informatique

	Avant-vente	Définition	Conception	Construction	Réception	Déploiement
Entrées		Expression du besoin client (cahier des charges, Appel d'offre...)				
Démarche		<ol style="list-style-type: none">1. Qualifier la demande du client2. Construire la réponse technique et commerciale<ol style="list-style-type: none">a. Organiser l'équipe de réponseb. Reformuler le besoin du client (identification des exigences dans un document) : <u>référentiel des exigences(product backlog)</u>c. Élaborer la solution (architecture matérielle + logicielle)d. Identifier les risquese. Estimer la <u>charge du projet en J/H</u> et définir l'équipe de projetf. Définir le planning de réalisation du projet (cycle de vie et <u>planning GANTT</u>)g. Remettre la réponse technique et commerciale3. En cas d'accord , signer un contrat				
Sortant		Réponse technique et commerciale, Contrat signé				

Les phases d'un projet informatique

“Cas d'un projet en ingénierie logicielle”

Avant-
vente

Définition

Conception

Construction

Réception

Déploiement

Entrée de phase	<p>Le client a accepté notre proposition</p> <p>Les éléments d'avant-vente sont disponibles (cahier de charges , référentiel des exigences...etc)</p>
Démarche	<p>1. Analyser les besoins : (diagrammes des cas d'utilisation, de séquences ...)</p> <p>a. Valider le référentiel d'exigences: Cela passe par les actions suivantes :</p> <ul style="list-style-type: none">i. Lire le cahier des charges du clientii. Prendre connaissance du référentiel d'exigences de l'avant-vente,iii. Faire une revue d'exigence pour détecter les exigences ambiguës, les oubliés..(on recueille et on analyse en détail les exigences du projet) <p>2. Organiser le projet : revoir les éléments du projet (livrables, ressources, charges, planning...)</p>
Sortant	Docs : <u>Spécifications fonctionnelles générales et planning détaillé (Ref CMMI)</u>

Les phases d'un projet informatique

“Cas d'un projet en ingénierie logicielle”

Avant-
vente

Définition

Conception

Construction

Réception

Déploiement

Entrée de phase	<ul style="list-style-type: none">■ Le document référentiel des exigences raffiné■ Spécifications fonctionnelles générales (document rédigé en sortie de la phase de définition)
Objectifs	<p>Permettre :</p> <ul style="list-style-type: none">■ Au client d'avoir une vision complète de l'application finale, tant d'un point de vue ergonomique et graphique que fonctionnel et technique■ Au équipes de développement d'avoir l'ensemble des éléments nécessaire pour construire l'application
Démarche	<ol style="list-style-type: none">1. Conception ergonomique et graphique2. Conception fonctionnelle3. Conception Technique

Les phases d'un projet informatique

“Cas d'un projet en ingénierie logicielle”

Avant-
vente

Définition

Conception

Construction

Réception

Déploiement

Conception graphique

Objectifs	<ul style="list-style-type: none">▪ Concevoir avec le client l'ergonomie de l'application à réaliser▪ Définir son identité visuelle par une création graphique▪ Produire une maquette décrivant de manière détaillée les écrans de l'application
Démarche	<ol style="list-style-type: none">1. <u>Définir le rubriquage</u> (l'arborescence des menus, cinématique de navigation)2. <u>Produire et valider le storyboard</u> : l'ensemble des écrans de l'application sans se soucier du design.3. <u>Produire et valider la charte graphique</u> : Définition de l'identité visuelle et design (couleurs, ...)4. <u>Réaliser le design des pages</u> : Créer la représentation graphique des pages de l'application à partir du storyboard, en tenant compte de la charte graphique (designer)5. <u>Produire la maquette</u>: déclinaison de la création graphique des pages dans la technologie retenue pour la couche présentation (développeur front)

Les phases d'un projet informatique

“Cas d'un projet en ingénierie logicielle”

Avant-
vente

Définition

Conception

Construction

Réception

Déploiement

Conception fonctionnelle

Démarche

1. Produire les spécifications fonctionnelles détaillées:

- Décrivent exhaustivement les fonctionnalités de l'application
- Elles seront le point d'entrée remis au développeur pour mener les tâches de réalisation puis de test unitaire
- Les spécifications détaillées permettent d'identifier l'ensemble des objets métiers de l'application et leurs attributs:
 - Modéliser les **objets métiers**
 - Créer des **diagrammes de classes** pour présenter les objets du domaine
 - Créer le **modèle de données pour prévoir la persistance**

Sortant

Document : Spécification fonctionnelles détaillées (Cf CMMI)

Les phases d'un projet informatique

“Cas d'un projet en ingénierie logicielle”

Avant-
vente

Définition

Conception

Construction

Réception

Déploiement

Conception technique

Démarche

1. **Définir l'architecture technique:** décrire l'architecture matérielle et logicielle de l'infrastructure cible hébergeant l'application. on définit :
 - Les machines sont mises en œuvre pour héberger le site.
 - Les versions des logiciels utilisés et leur positionnement sur les machines (SGBD , serveur d'application,...)
1. **Définir l'architecture applicative :**
 - **Les couches applicatives** mises en oeuvre
 - Les services transversaux (logs, authentification, persistance...)
 - Introduit la manière d'implémenter chacune des couches (Savoir quel framework/composant sera utilisé dans les différentes couches)
1. **Réaliser la conception technique détaillée**
 - **On sélectionne 2 à 3 fonctionnalités** représentatives de l'architecture et on définit leur implémentation sur la totalité de l'architecture. Il sert donc d'exemple de codage de l'ensemble des couches et services applicatifs.

Sortant

- ✓ Document de Spécifications techniques détaillées

Les phases d'un projet informatique

“Cas d'un projet en ingénierie logicielle”

Avant-
vente

Définition

Conception

Construction

Réception

Déploiement

Entrée de phase	<ul style="list-style-type: none">▪ Livrables de conception validés
Objectifs	<ul style="list-style-type: none">▪ Développer l'application selon les spécifications fonctionnelles et techniques, ainsi que d'évaluer sa qualité et la documenter.
Démarche	<ol style="list-style-type: none">1. Réalisation des développements2. Développement et tests unitaires3. Intégration continue (branche partagée “main” github)4. Recette interne (tests d'intégration)5. Correction des anomalies

Les phases d'un projet informatique

“Cas d'un projet en ingénierie logicielle”

Avant-
vente

Définition

Conception

Construction

Réception

Déploiement

Entrée de phase	<ul style="list-style-type: none">La solution a été testée / vérifiée en interne à la fin de constructionLa solution est installée dans l'environnement de recette(pré-production)
Objectifs	<ul style="list-style-type: none">Vérifier que le projet a été développé conformément aux spécifications fonctionnelles et aux spécifications techniques définies en phase de conceptionVérifier l'intégration technique de l'application avec le SI du client
Démarche	<ol style="list-style-type: none">livraison de la solution chez le clientInstallation de la solution et accompagnement du client pour la recetteFormation des utilisateursDocumentation de l'application (fonctionnelle, exploitation)Application du workflow de traitement des anomalies

Les phases d'un projet informatique

“Cas d'un projet en ingénierie logicielle”

Avant-
vente

Définition

Conception

Construction

Réception

Déploiement

Entrée de phase	<ul style="list-style-type: none">La solution est validée par le client,
Démarche	<ul style="list-style-type: none">Déploiement de la solution en productionAssistance à l'installation de l'environnement de productionPackaging du projetFormation des utilisateursTransfert de compétences aux équipes techniquesClôture du projetBilan de projetArchivage du projet et capitalisation

Phases d'un projet Informatique

Autres exemples (Cas d'un Projet en réseaux informatique)

Projet d'installation du réseau informatique à la FSO

- 1. Définition** des besoins (cahier de charges) : blocs à couvrir(départements , administration, bibliothèque, ...) , couverture wifi , liaisons fibre, nombre d'étages et de machines par étage / bloc, plateforme de supervision , pare-feu , VLANs pour séparer entre les réseaux (étudiants , profs, admin), virtualisation côté serveurs ...
- 2. Conception** de l'architecture réseau : schéma de l'architecture globale , par bloc ...
- 3. Construction** : Achat du logiciel + matériel (câblage , switchs, serveurs, licences OS , logiciel de virtualisation , ...) , configuration, installation et déploiement
- 4. Reception** : test de la nouvelle architecture réseaux par le client (FSO)
- 5. Déploiement** : migration et mise en production

Phases d'un projet Informatique

Autres exemples : Projet IDS Basé sur l'IA

- **Définition :**
Collecter, nettoyer et préparer les données (dataset) issues des dispositifs IoT.
- **Conception :**
Choisir le modèle d'IA approprié (Random Forest , SVM, LSTM).
- **Construction :**
Diviser les données en ensembles d'entraînement, de validation et de test, **entraîner le modèle** et affiner ses hyperparamètres pour optimiser la détection des intrusions.
- **Réception :**
Tester le modèle en conditions réelles, mesurer ses performances et recueillir les retours des équipes de sécurité pour confirmer son efficacité.
- **Déploiement :**
Intégrer le système dans l'infrastructure IoT pour générer des alertes en temps réel, avec un suivi continu et des mises à jour régulières du modèle.

Cycles de vie d'un projet



- Cycle de vie en cascade
- Cycle de vie en V
- Cycle itératif

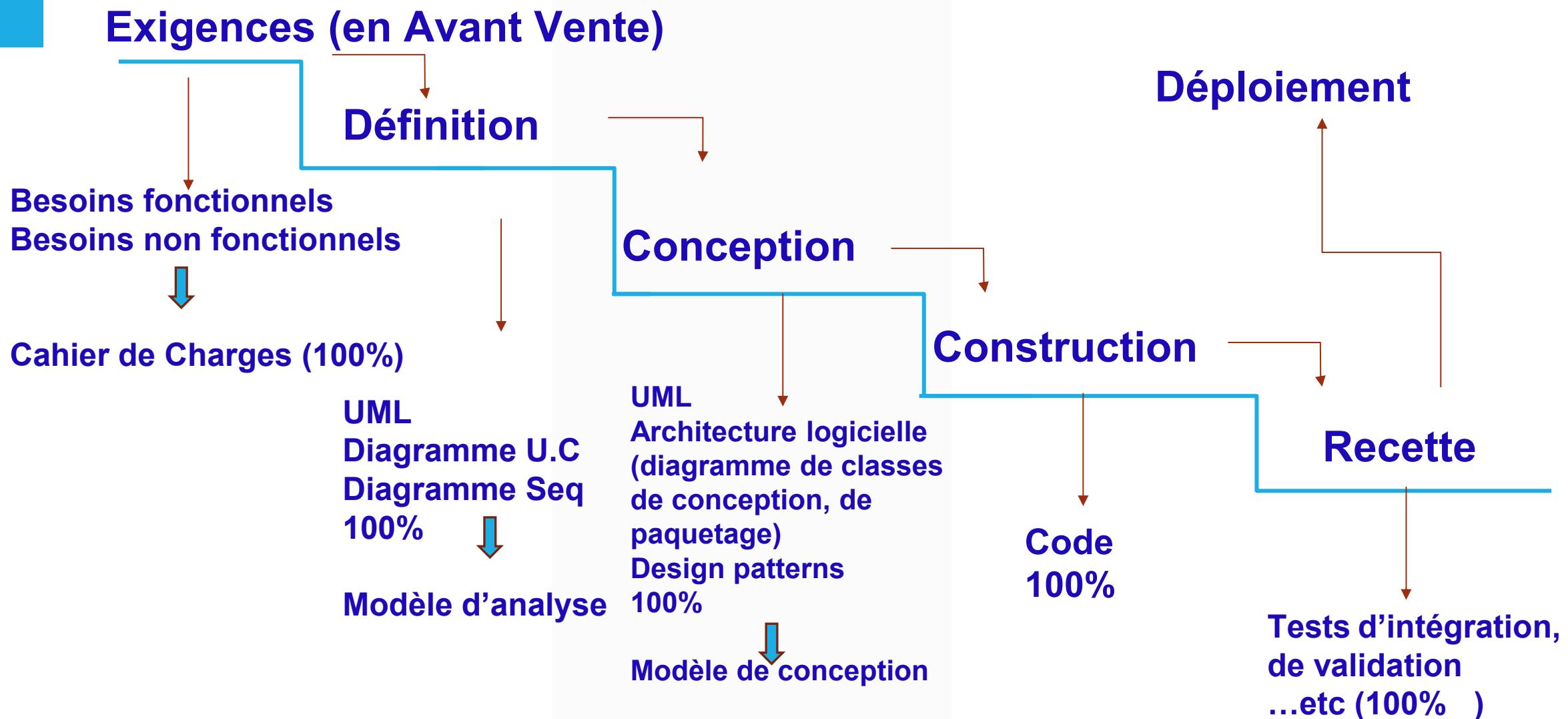
Cycle de vie du projet

- Le cycle de vie correspondant au **déroulement temporel des phases** et activités d'un projet, depuis son lancement jusqu'à la fin
 - ❖ Définir le cycle de vie est une **activité de la phase de l'avant vente**
 - ❖ Principaux cycles de vie:
 - 1) **Cycle de vie en cascade**
 - 2) **Cycle de vie en V**
 - 3) **Cycle itératif**

Cycle de vie en cascade

- ❖ **Toutes les activités** sont menées **séquentiellement**, la preuve de fonctionnement du système n'est apportée qu'à la fin complète du projet
- ❖ **Nécessite que les besoins** soient **globalement stables**
- ❖ Il faut alors que le délai du projet soit limité (< 5 mois) pour éviter de nombreux changements entre début et fin

Le Cycle de vie en cascade



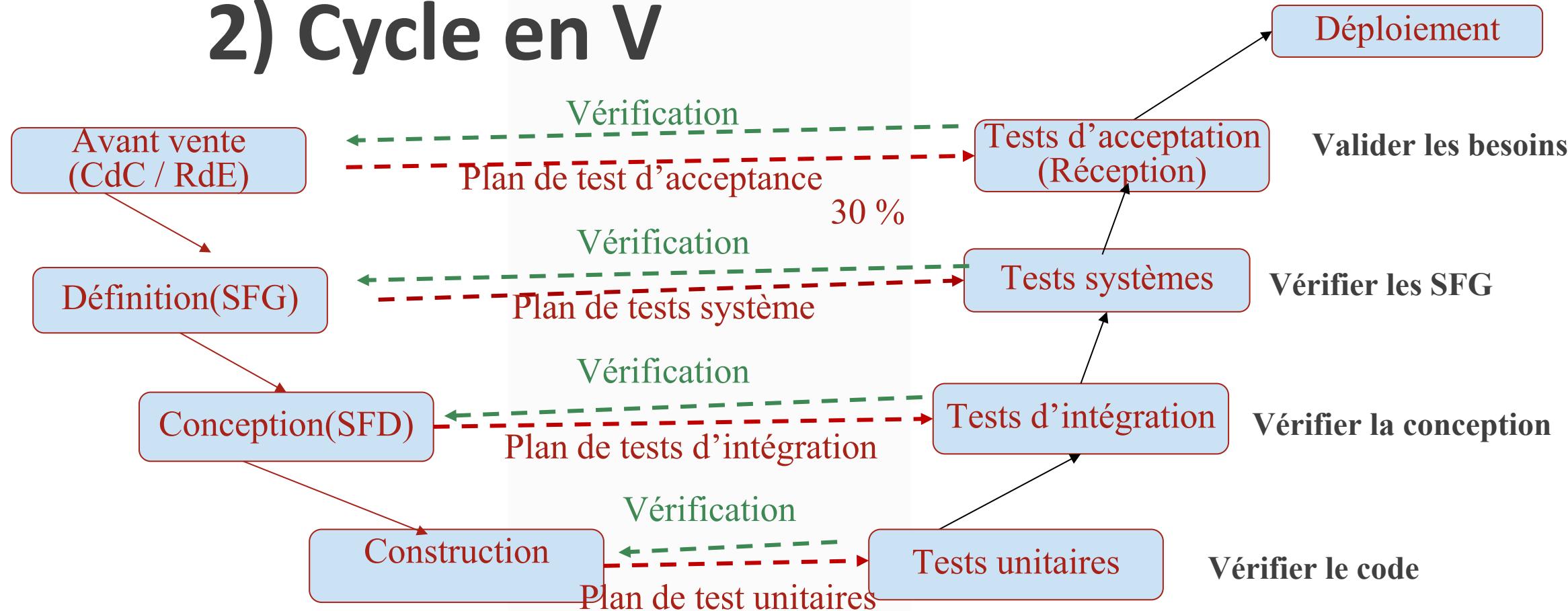
Critiques du cycle de vie en cascade 1/2

- ❖ Le modèle en cascade se base sur des exigences exprimées en début de projet. Toutefois les exigences et besoins peuvent se montrer incomplets ou de qualité insuffisante (**ambiguité, incohérence, etc.**).
- ❖ De plus, le **client peut ne pas être pleinement conscient de ses exigences** avant d'avoir vu le logiciel fonctionner.
- ❖ Ceci peut conduire à revoir la conception, re-développer une partie du logiciel, et re-tester le produit et donc augmenter les coûts. C'est pourquoi le modèle en cascade est particulièrement adapté à des petits projets simples, pour lesquels nous avons la quasi-certitude que les besoins ou exigences n'évolueront pas en cours de projet.

Critiques du cycle de vie en cascade 2/2

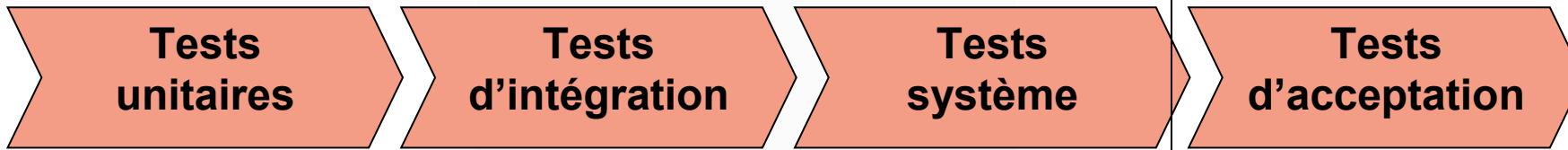
- ❖ La structuration des phases préconisée par le modèle en cascade **ne favorise pas suffisamment l'implication du client tout au long du projet**, et **déourage** la prise en compte des **changements**. Ce dernier point explique l'émergence d'une **approche incrémentale** (cycle de vie itératif)

2) Cycle en V



- ❖ La particularité par rapport du modèle en cascade c'est l'anticipation des tests sans attendre le code (établir des plans de test)
 - 30 % des tests des fonctionnalités => revenir aux phases en vis-à-vis pour rectifier
- ❖ Utile pour les projets de petite taille, avec un planning serré

Tests



**Plan de vérification
par les développeurs / testeurs (QA)**

**Plan de recette
(test & validation par le client)**

Unitaire	<p>L'objectif d'un test « unitaire » est de vérifier le bon fonctionnement du composant considéré.</p> <p>Rédaction des cas de tests <input type="checkbox"/> exécution des cas de tests</p>
Intégration	<p>L'objectif d'un test d' « intégration » est de vérifier la bonne interaction de plusieurs composants entre eux et de vérifier que le comportement des composants tiers est bien celui attendu.</p> <p>Rédaction des cas de tests <input type="checkbox"/> exécution des cas de tests</p>
Système	<p>L'objectif d'un test de niveau « système » est de vérifier le bon fonctionnement du système dans sa globalité(effectués en interne par le CP, CPT, QA) : test de fonctionnalités, de performance, de sécurité..etc</p> <p>Rédaction des cas de tests <input type="checkbox"/> exécution des cas de tests</p>
Tests d'acceptation	<p>Les tests d'acceptation sont généralement effectués pour vérifier si le livrable satisfait les besoins définis par le client (effectués par le client)</p> <p>Rédaction des cas de tests <input type="checkbox"/> exécution des cas de tests</p>

Tableau Comparatif - Tests

Type de Test	Qui prépare le plan de test ?	Quand préparer le plan de test ? (Phase du cycle en V)	Qui exécute le test ?	Quand exécuter le test ? (Phase du cycle en V)	Objectif du test	Exemple de scénario testé
Test Unitaire	Développeurs	Construction (Développement)	Développeurs (automatisé)	Tests Unitaires (après codage)	Vérifier que chaque fonction ou module fonctionne indépendamment.	Tester si une fonction de calcul de remise donne le bon prix.
Test d'Intégration	Développeurs & QA (Quality Assurance)	Conception (SFD - Spécification Fonctionnelle Détailée)	Développeurs & QA	Tests d'Intégration (après les tests unitaires)	Vérifier que les modules communiquent correctement entre eux.	Vérifier qu'un paiement validé déclenche bien une commande en base de données.
Test Système	Équipe QA (Testeurs, Ingénieurs Test)	Définition (SFG - Spécification Fonctionnelle Générale)	Équipe QA (manuels ou automatisés)	Tests Systèmes (après les tests d'intégration)	Vérifier que l'ensemble du système fonctionne correctement (performance, sécurité, compatibilité).	Tester si un site e-commerce supporte 10 000 utilisateurs simultanés sans crash.
Test d'Acceptation	Clients, utilisateurs finaux, Product Owner, MOA (Maîtrise d'Ouvrage)	Avant-Vente (CdC / RdE - Cahier des Charges & Recueil des Exigences)	Clients, utilisateurs finaux, parties prenantes métier	Tests d'Acceptation (Réception - avant déploiement)	Valider que le logiciel répond aux besoins métier et est prêt pour la mise en production.	Vérifier si un utilisateur peut acheter un produit en moins de 5 clics .

Plan de test - exemple d'un cas de test (TU)

Cas de test unitaire	Ajout d'un nouveau produit
Prérequis	L'utilisateur doit être connecté en tant que admin
Actions à effectuer	Cliquer sur Gestion du catalogue des produits dans le menu Cliquer sur Produits Cliquer sur Ajouter + Remplir les champs (réf, prix, desc, catégorie...) Cliquer sur valider (sinon annuler pour réinitialiser)
Résultats attendus	Message de confirmation de réussite de l'opération Page liste de produits actualisée Produit ajouté dans la base de données ...

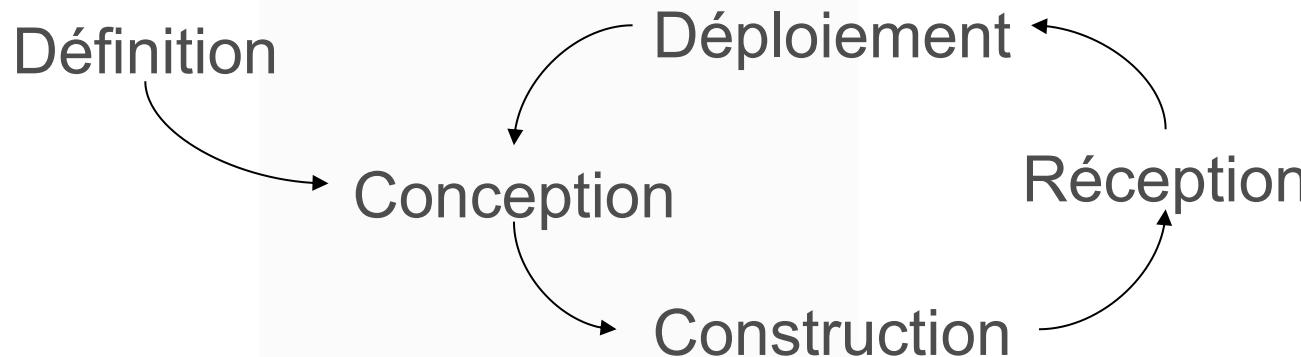
Les avantages du cycle en V

- ❖ Les **chances de réussite** sont plus **élevées** que pour la méthode en cascade grâce à l'**élaboration d'un plan de test** dès le début du cycle de développement.
- ❖ Il réduit le nombre de retouches grâce à la détection précoce des défauts (bugs) et des problèmes.

3) Cycle de vie itératif et incrémental

- ❖ Le système est conçu et construit **progressivement**, par **succession d'itérations** apportant chacune un raffinement sur le contenu de l'application
- ❖ Permet de mieux **gérer les changements** et les situations où le besoin de l'utilisateur est peu formalisé, complexe ou les projets s'étalent sur une longue durée
- ❖ Permet de **réduire les risques** car l'utilisateur final s'approprie l'application progressivement

3) Cycle de vie itératif et incrémental



- ❖ Il y a généralement une seule phase de définition
- ❖ Chaque itération est une séquence **conception/construction/réception/déploiement**
- ❖ Les **itérations sont incrémentales**, enrichissant le périmètre du projet progressivement
- ❖ La première itération est la plus difficile à produire
 - **Elle intègre la formation des équipes, la mise en place de l'architecture technique (environnement de développement , de test et de production)**

Critères de choix du cycle de vie

Choix du cycle de vie			
Type de projet	Petit projet	Moyen projet	Grand projet
Complexité	Simple	de complexité moyenne	complexe
Besoins client	Clair et stable	Relativement stable	Peu clairs et instables
risques	Risques faibles	Risques fonctionnels élevés	Risques élevés dans tous les niveaux
Charge ou durée du projet	< 150 jh (jours/homme)	> 150 jh et <500 jH	> 500 jh
Cycle recommandé	Cycle en cascade / V (approche prédictive)	Cycle semi-itératif sur les phases construction / réception (approche hybride)	Cycle itératif complet (approche agile)