

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Mila

Département Math & Informatique

Génie logiciel II

CHAPITRE 2.

Processus de développement logiciel (Cycle de vie d'un logiciel)

HEDJAZ Sabrina

2019-2020

Processus de développement logiciel (Cycle de vie)

Cycle de vie d'un logiciel:

- ❑ Le cycle de vie du logiciel modélise l'enchaînement des différentes activités du processus technique de développement du logiciel.

Une activité comprend: des tâches, des contraintes, des ressources, une façon d'être réalisée.

- ❑ Les grandes activités sont:
 - ✓ Analyse des besoins
 - ✓ spécification globale
 - ✓ Conception architecturale et détaillé
 - ✓ Programmation
 - ✓ Gestion de configuration et d'intégration
 - ✓ Validation et vérification
 - ✓ Livraison et maintenance

Processus de développement logiciel (Cycle de vie)

1. Composant d'un cycle de vie d'un logiciel:

1.1. Analyse des besoins:

- ❑ But: Eviter de développer un logiciel non adéquat. On va étudier le domaine d'application ainsi que l'état actuel et futur de l'environnement du système afin d'en déterminer: Les frontières, Le Rôle, Les ressources disponibles et requises, Les contraintes d'utilisation et performance ... etc.

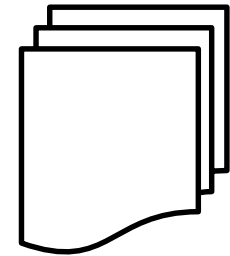


- Experts du domaines d'application
- Futur utilisateurs du système



Analyse des besoins

- Entretien
- Questionnaire
- Observation de l'existence
- Etude de situation similaire



- Cahier des charges (CC)
- Manuel d'utilisation préliminaire

Processus de développement logiciel (Cycle de vie)

1. Composant d'un cycle de vie d'un logiciel:

1.2. Spécification globale :

- ❑ But: Etablir une première description du futur système. pour produire une description de ce que doit faire le système mais sans préciser comment il le fait



- Cahier des charges (CC)
- Manuel d'utilisation préliminaire
- Diagramme de flot de données
- Diagramme de contexte
- Diagramme d'état transition
- Réseaux de pétri et les grafcet
- Diagramme des méthodes O.O ...
- Modèle E/A
- Cahier des charges fonctionnel
- Technique et faisabilité informatique

Processus de développement logiciel (Cycle de vie)

1. Composant d'un cycle de vie d'un logiciel:

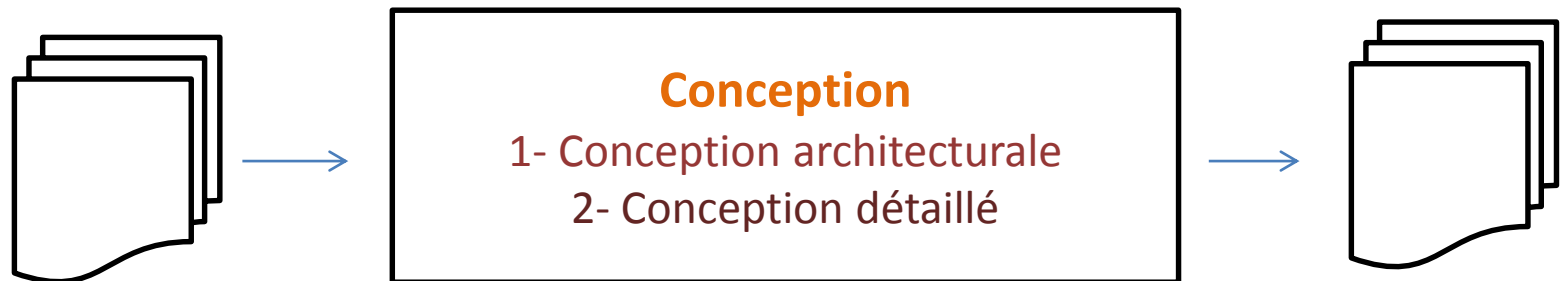
1.3. Conception:

- ❑ But: Enrichir la description du logiciel de détails d'implémentation afin d'aboutir à une description très proche d'un programme (décrire le comment).
- ✓ La conception architecturale (ou conception globale) a pour but de décomposer le logiciel en composants plus simples, définis par leurs interfaces et leurs fonctions (les services qu'ils rendent).
- ✓ La conception détaillée fournit pour chaque composant une description de la manière dont les fonctions ou les services sont réalisés : algorithmes, représentation des données.

Processus de développement logiciel (Cycle de vie)

1. Composant d'un cycle de vie d'un logiciel:

1.3. Conception:



- Cahier des charges fonctionnel
- Technique et faisabilité informatique
- Conception fonctionnelle et conception orienté objet
- Pour chaque composant, écrire les algorithmes
- Dossier de conception: (Architecture et Algorithme)

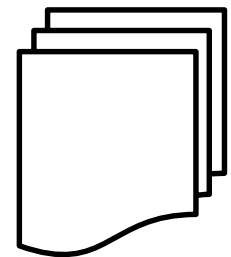
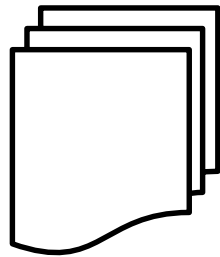
Processus de développement logiciel (Cycle de vie)

1. Composant d'un cycle de vie d'un logiciel:

1.4. Programmation:

Implantation de la solution conçue

- ✓ Choix de l'environnement de développement, du/des langage(s) de programmation, de normes de développement...



- Dossier de conception:
(Architecture et
Algorithme)

- Code source
- Manuel final
d'utilisateur
- Documentations

Processus de développement logiciel (Cycle de vie)

1. Composant d'un cycle de vie d'un logiciel:

1.5. Gestion de configuration et intégration:

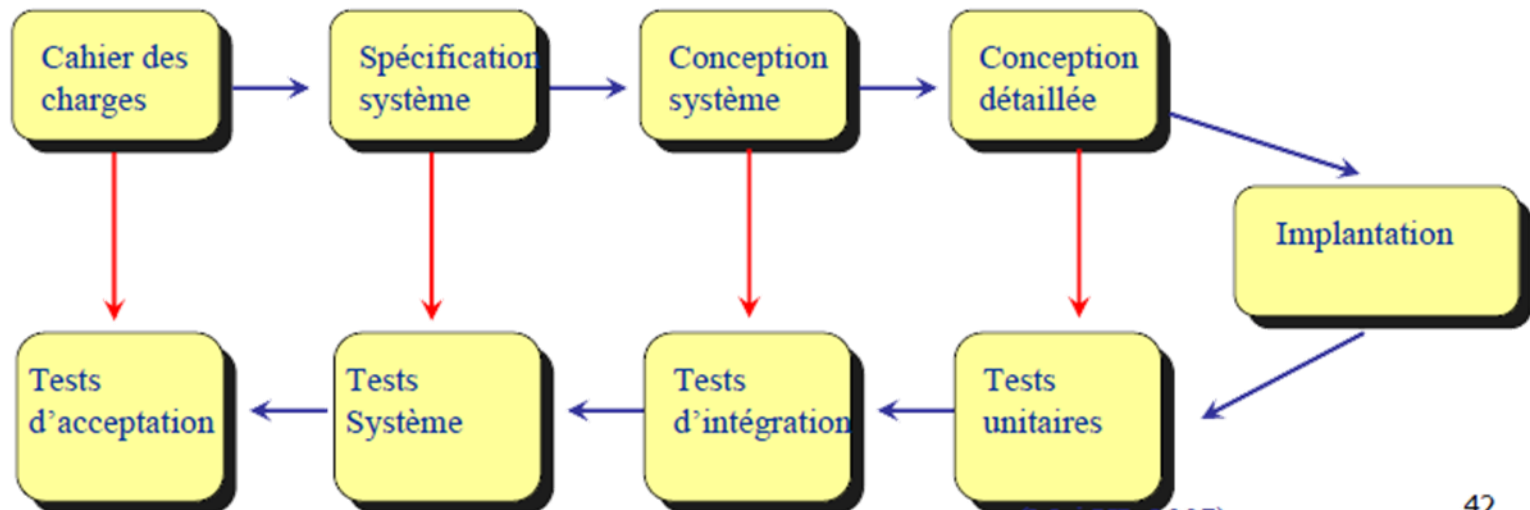
- ✓ La gestion de configurations a pour but de maîtriser l'évolution et la mise à jour des composants tout au long du processus de développement.
- ✓ L'intégration a pour but de réaliser un ou plusieurs systèmes exécutables à partir des composants (Combiner les composants).

Processus de développement logiciel (Cycle de vie)

1. Composant d'un cycle de vie d'un logiciel:

1.7. Validation et vérification:

- ✓ La **validation** a pour but de répondre à la délicate question : a-t-on décrit le bon système, celui qui répond à l'attente des utilisateurs.
- ✓ La **vérification** répond à la question : le développement est-il correct par rapport à la spécification globale ? Ce qui consiste à s'assurer que les description successives et le logiciel lui-même satisfont la spécification.



Processus de développement logiciel (Cycle de vie)

1. Composant d'un cycle de vie d'un logiciel:

1.8. Maintenance

- ✓ Il s'agit d'apporter des **modifications** à un logiciel existant. C'est la phase la **plus coûteuse** (70% du coût total)

- ✓ Types:

Maintenance corrective, Maintenance perfective, Maintenance adaptative.

Processus de développement logiciel (Cycle de vie)

1. Composant d'un cycle de vie d'un logiciel:

1.8. 1, Maintenance corrective

- Corriger les erreurs : défauts d'utilité, d'utilisabilité, de fiabilité...
 - Identifier la défaillance, le fonctionnement
 - Localiser la partie du code responsable
- Attention
 - La plupart des corrections introduisent de nouvelles erreurs
 - Les coûts de correction augmentent exponentiellement avec le délai de détection
 - Corriger et estimer l'impact d'une modification
- La maintenance corrective donne lieu à de nouvelles livraisons (release)

Processus de développement logiciel (Cycle de vie)

1. Composant d'un cycle de vie d'un logiciel:

1.8. 1, Maintenance adaptative

- Ajuster le logiciel pour qu'il continue à remplir son rôle compte tenu de l'évolution des
 - Environnements d'exécution
 - Fonctions à satisfaire
 - Conditions d'utilisation
- Ex : changement de SGBD, de machine, de taux de TVA, an 2000, euro...

Processus de développement logiciel (Cycle de vie)

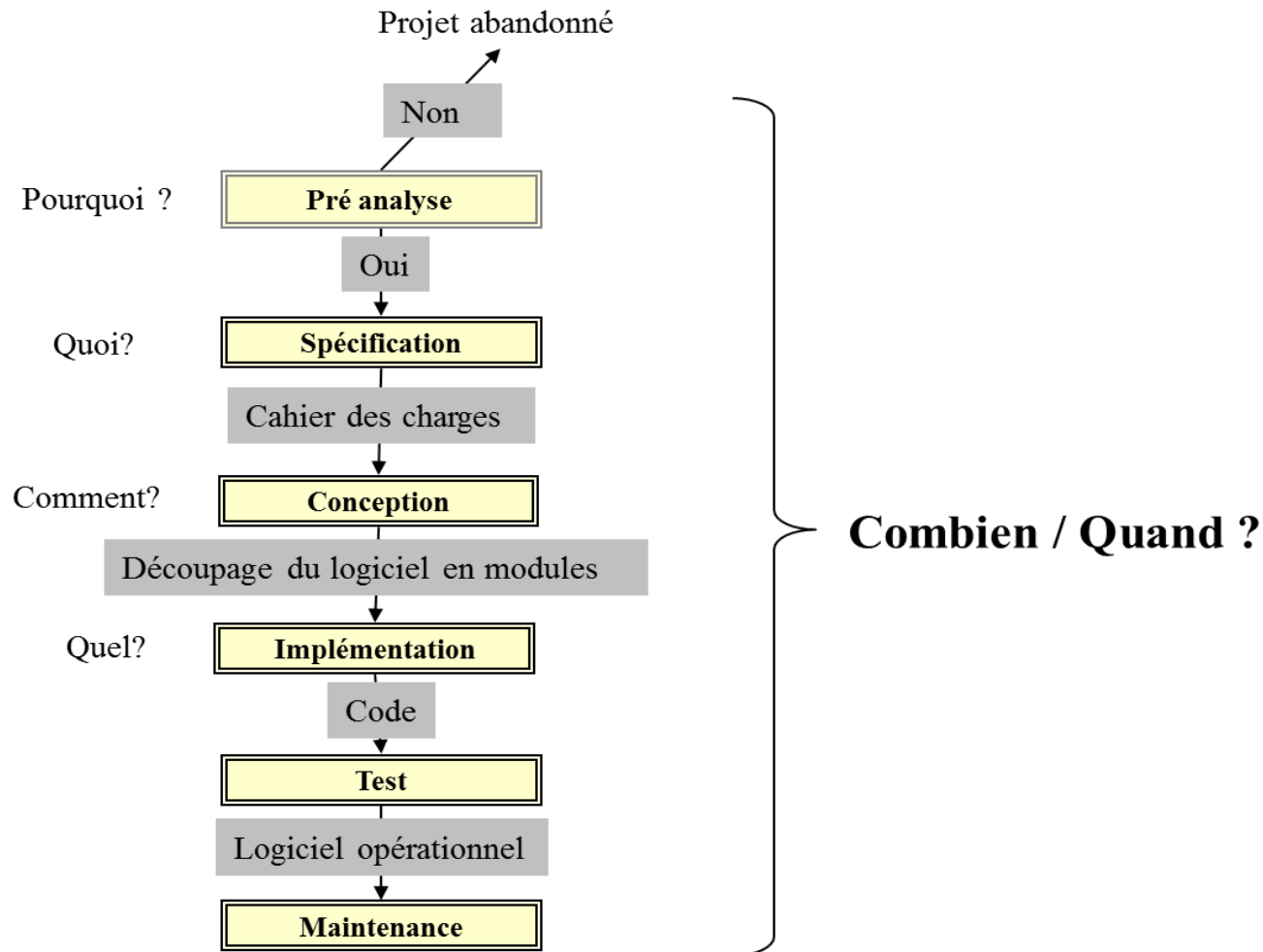
1. Composant d'un cycle de vie d'un logiciel:

1.8. 1, Maintenance perfective, extension

- Accroître/améliorer les possibilités du logiciel
- Ex : les services offerts, l'interface utilisateur, les performances...
- Donne lieu à de nouvelles versions

Processus de développement logiciel (Cycle de vie)

1. Composant d'un cycle de vie d'un logiciel:



Processus de développement logiciel (Cycle de vie)

2. Documents

✓ Manuel utilisateur final	Implémentation
✓ Dossier de conception architecturale	Conception
✓ Code source	Implémentation
✓ Cahier des charges	Analyse des besoins
✓ Manuel utilisateur préliminaire	Analyse des besoins
✓ Dossier de conception détaillée	Conception
✓ Rapport des tests	Tests
✓ Documentation	Implémentation
✓ Cahier des charges fonctionnel	Spécification

Processus de développement logiciel (Cycle de vie)

3. Modèle de cycle de vie logiciel:

☐ Modèles linéaires

- ✓ modèle en cascade
- ✓ modèle en V
- ✓ ...

☐ Modèles non linéaires

- ✓ prototypage
- ✓ modèles incrémentaux
- ✓ modèle en spirale
- ✓ Modèles unifiés
- ✓ ...

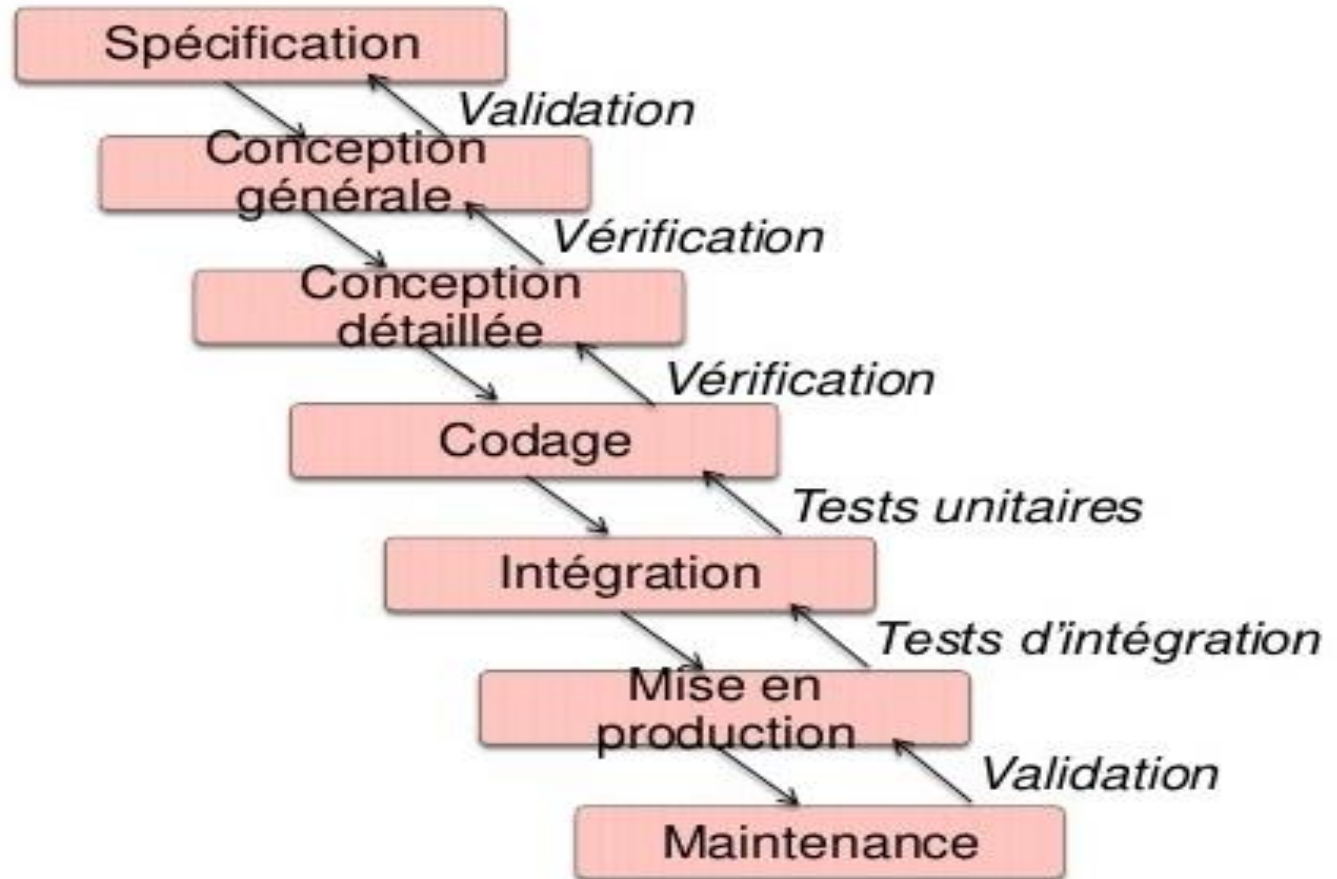
Processus de développement logiciel (Cycle de vie)

3. 1. Modèle en cascade

- ☐ Date des années 70 mais reste pertinent (Royce)
- ☐ Modèle linéaire avec phases séquentielles
- ☐ Vérification de chaque phase avant de passer à la suivante
- ☐ On ne peut passer à l'étape suivante tant la précédente n'est pas terminée
- ☐ Production de documents à l'issue de chaque phase
- ☐ la modification d'une étape a un impact important sur les étapes à venir

Processus de développement logiciel (Cycle de vie)

3. 1. Modèle en cascade



Processus de développement logiciel (Cycle de vie)

3. 1. Avantages et inconvénients

❑ Avantages

- ✓ Le planning est établi à l'avance et le chef du projet sait précisément ce qui va lui être livré et quand il pourra en prendre livraison

❑ Inconvénients

- ✓ Modèle trop séquentiel c.à.d. dure trop longtemps
- ✓ Validation trop tardive (remise en question coûteuse des phases précédentes)
- ✓ Sensibilité à l'arrivée de nouvelles exigences (refaire toutes les étapes)
- ✓ Bien adapté lorsque les besoins sont clairement identifiés et stables,

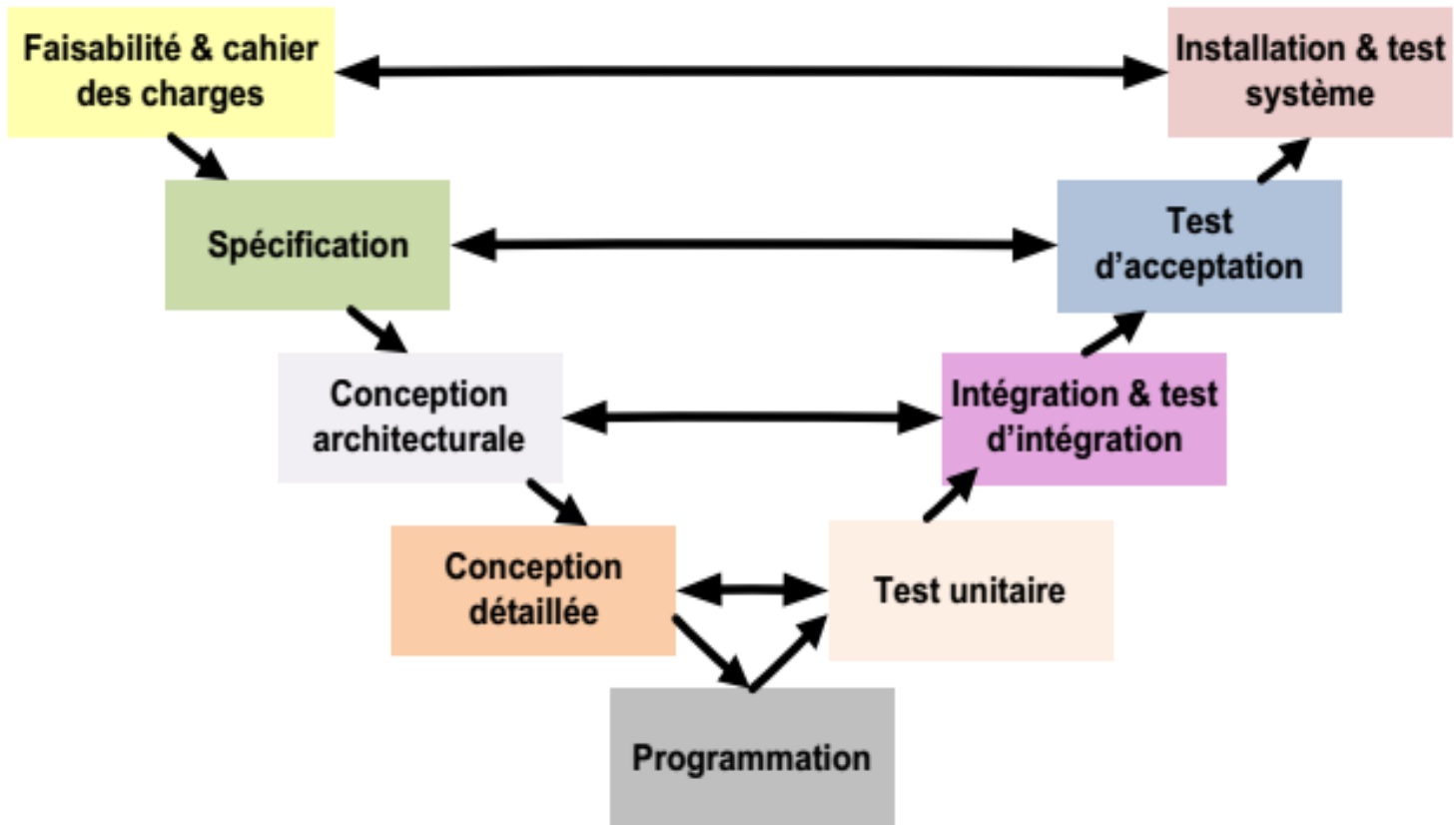
3. 2. Modèle en V

A ce jour, le cycle en V reste le cycle de vie le plus utilisé. C'est un cycle de vie **orienté test** :

- A chaque **activité créative** (spécification, conception et codage) correspond une **activité de vérification** (validation, intégration, tests unitaires).
- Chaque phase en amont prépare la phase correspondante de vérification (la vérification est prise en compte au moment même de la création).

Processus de développement logiciel (Cycle de vie)

3. 2. Modèle en V



Processus de développement logiciel (Cycle de vie)

3. 2. Avantages et inconvénients

Avantages

- La préparation des dernières phases (validation-vérification) par les premières (construction du logiciel), permet d'éviter d'énoncer une propriété qu'il est impossible de vérifier objectivement après la réalisation.

Inconvénients

- Construit-on le bon logiciel ? Le logiciel est **utilisé très (trop) tard**.
 - Il faut attendre longtemps pour savoir si on a construit le bon logiciel.
 - Difficile d'impliquer les utilisateurs lorsque qu'un logiciel utilisable n'est disponible qu'à la dernière phase
- **Idéal** quand les besoins sont bien connus, quand l'analyse et la conception sont claires

Processus de développement logiciel (Cycle de vie)

3. 3. Prototypage évolutif et jetable

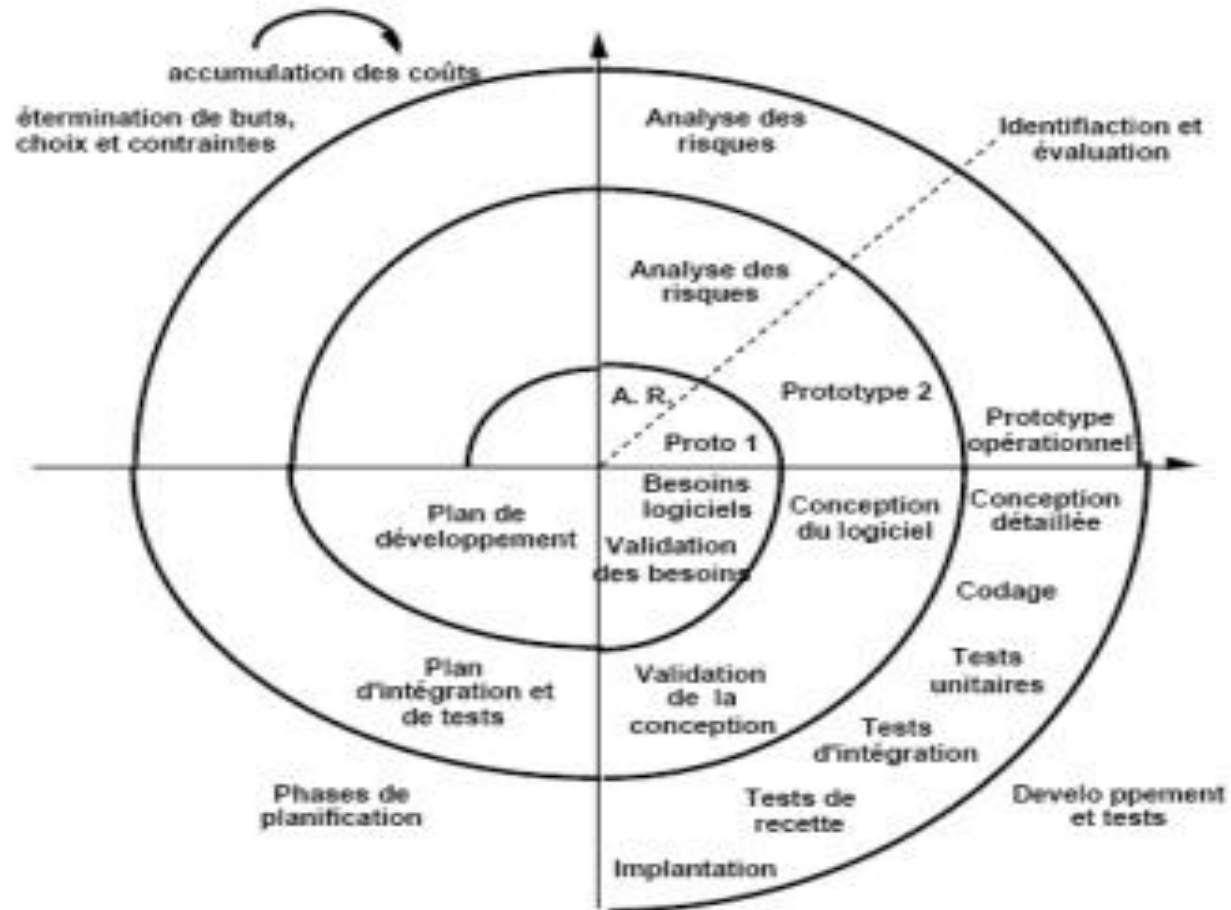
- (Prototype jetable) : squelette du logiciel qui n'est créé que dans un but et dans une phase particulière du développement
- Si gardé, alors s'appelle (prototype évolutif)
 - Première version du prototype = embryon du produit final
 - On itère jusqu'au produit final

3. 3. Prototypage avantages et inconvénients

- **Avantage :**
 - Les efforts consacrés au développement d'un prototype sont le plus souvent compensés par ceux gagnés à ne pas développer de fonctions inutiles
- **Mais :**
 - Les décisions rapides sont rarement de bonnes décisions
 - Le prototype évolutif donne-t-il le produit demandé ?

Processus de développement logiciel (Cycle de vie)

3. 4. spirale



Processus de développement logiciel (Cycle de vie)

3. 4. spirale (cycle de vie)

Chaque cycle de la spirale se déroule en quatre phases :

- ① détermination, à partir des résultats des cycles précédents, ou de l'analyse préliminaire des besoins, des objectifs du cycle, des alternatives pour les atteindre et des contraintes ;
- ② analyse des risques, évaluation des alternatives et, éventuellement maquettage ;
- ③ développement et vérification de la solution retenue, un modèle « classique » (cascade ou en V) peut être utilisé ici ;
- ④ revue des résultats et vérification du cycle suivant.

Processus de développement logiciel (Cycle de vie)

3. 4. spirale (Analyse des risques)

Risques technologiques

- exigences démesurées par rapport à la technologie
- l'incompréhension des fondements de la technologie
- changement de technologie en cours de route
- ...

Risques liés au processus

- gestion projet mauvaise ou absente
- calendrier et budget irréalistes
- calendrier abandonné sous la pression des clients
- développement de fonctions inappropriées
- ...

Risques humains

- défaillance du personnel
- surestimation des compétences
- travailleur solitaire
- manque de motivation

Processus de développement logiciel (Cycle de vie)

3. 4. spirale (Analyse des risques)

Avantages

- Permet d'établir le modèle de développement le plus adéquat en regard du risque.
- **Méta-modèle**: tous les autres modèles de développement constituent une variante de la spirale!

Inconvénients

- ❑ A besoin de **compétences dans l'évaluation approfondie des incertitudes et des risques** associés au projet et leur réduction.
- ❑ Évaluer les risques impliqués dans le projet peut prendre jusqu'à le **coût et il peut être plus élevé que le coût de la construction du système.**

Processus de développement logiciel (Cycle de vie)

4. Modèle de processus logiciel:

- ❑ On peut vouloir représenter le processus de développement (et ses parties) de manière plus fine.
- ❑ Il est nécessaire de se doter d'un formalisme pour représenter le processus de développement.
 - ✓ Pour ordonner les tâches.
 - ✓ Pour déterminer les échanges d'information entre les différentes tâches.
 - ✓ Pour permettre aux jeunes recrues de mieux travailler.

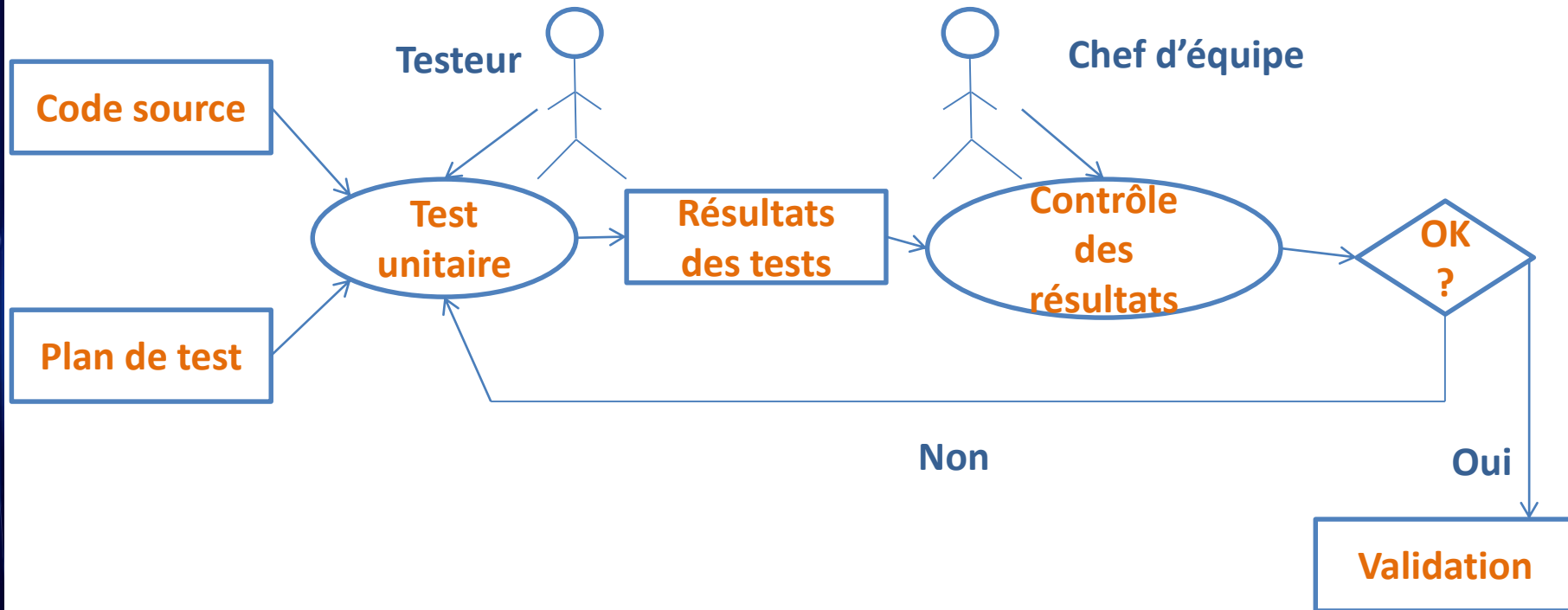
Processus de développement logiciel (Cycle de vie)

4. Modèle de processus logiciel:

- ❑ Un modèle de processus logiciels décrit
 - ✓ Les tâches
 - ✓ Les artefacts (fichiers, documents, données...)
 - ✓ Les auteurs
 - ✓ Les décisions (facultatif)
- ❑ Règles à observer
 - ✓ Deux tâches doivent être séparées par un artefact
 - ✓ Une tâche ne peut être exécutée tant que ses artefacts d'entrée n'existent pas
 - ✓ Il doit y avoir un trajet depuis chaque tâche jusqu'à la tâche de fin
 - ✓ Il doit y avoir au moins une tâche de début et une de fin

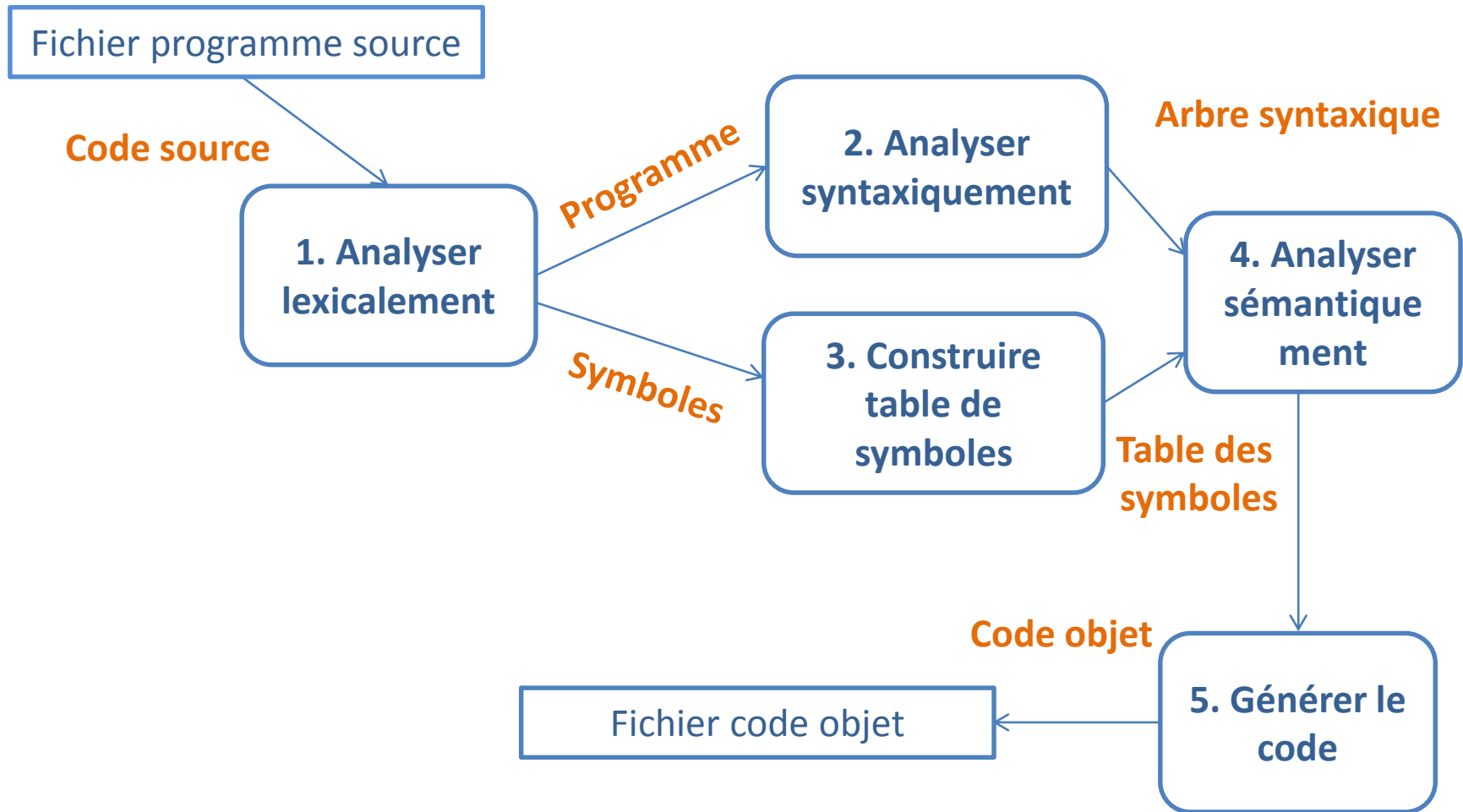
Processus de développement logiciel (Cycle de vie)

4.1. Exemple de processus logiciel

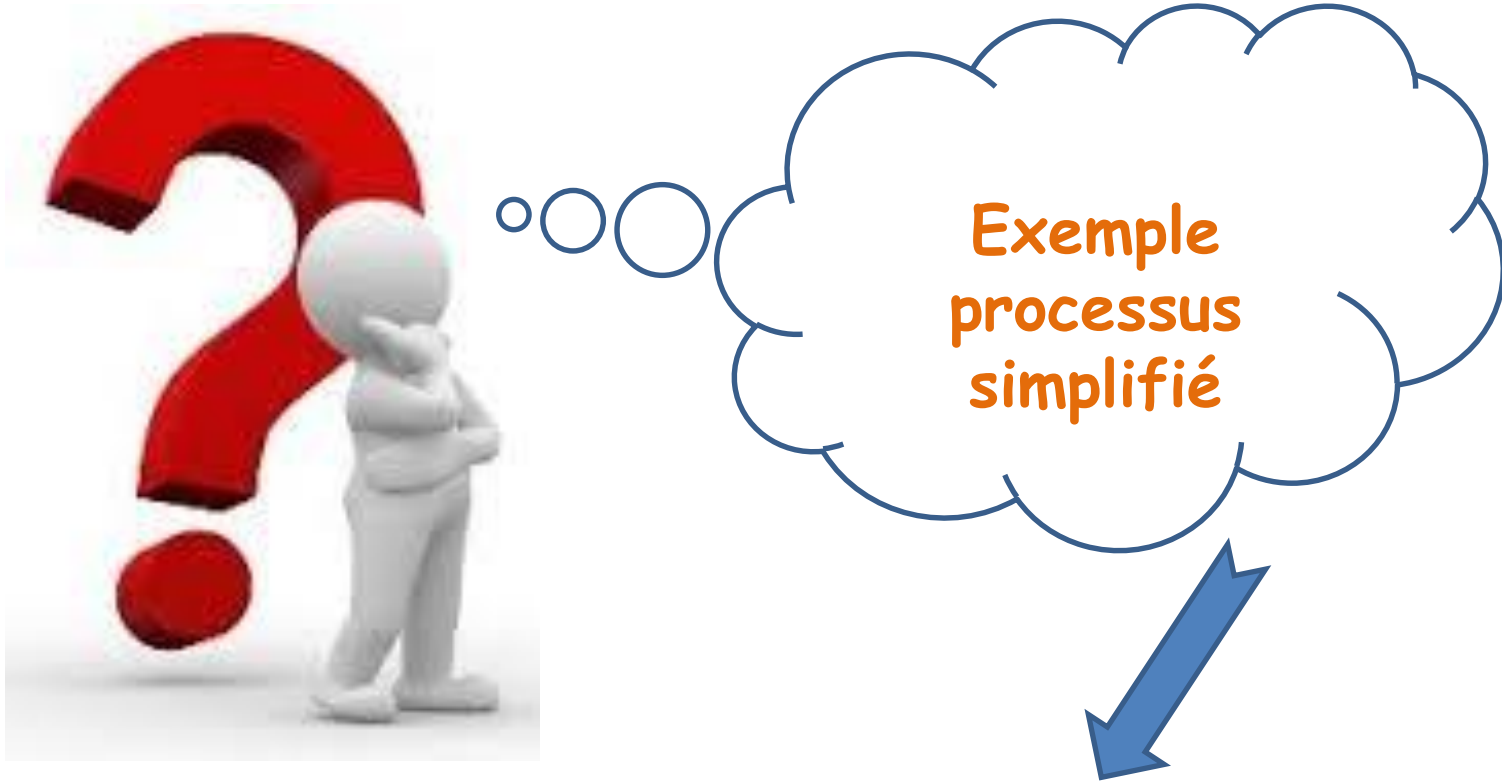


Processus de développement logiciel (Cycle de vie)

4.2. Exemple de DFD (compilateur):



Processus de développement logiciel (Cycle de vie)



Chapitre 3