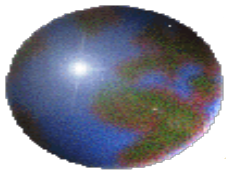


Méthode MERISE

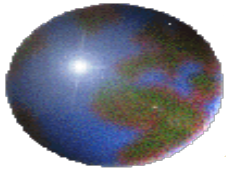
Chap. 2. Principes fondamentaux de MERISE: Le cycle de vie

Dr. Coulibaly Tiékoura

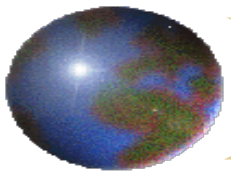


Plan

- ✚ 2.1. Cycles de développement d'un système d'information
- ✚ 2.2. Le Cycle de vie
- ✚ 2.3. Les différents modèles de cycles de vie
 - ✚ Cycle de vie en cascade
 - ✚ Cycle de vie en V
 - ✚ Cycle de vie en spirale

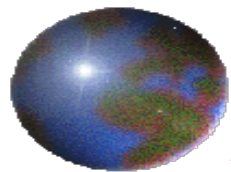


2.1. Cycles de développement d'un système d'information



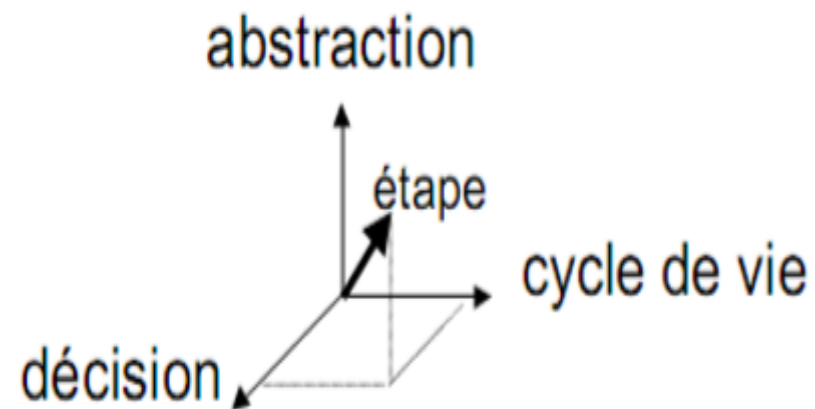
2.1. Cycles de développement d'un système d'information

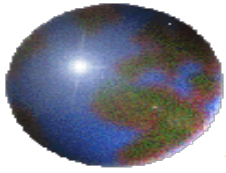
- ✚ La démarche de développement d'un SI doit être conduite suivant trois axes appelés **cycles** dont le degré de prise en compte par les différentes méthodes oriente le choix de l'une d'entre elles en fonction des objectifs de l'étude.
- ✚ Ces 3 cycles sont:
 - ✚ **Le cycle de vie:** la démarche (comment enchaîner les étapes?)
 - ✚ **Le cycle d'abstraction:** le raisonnement (quels outils permettent de les mener?)
 - ✚ **Le cycle de décision:** la maîtrise (quelles décisions sont à prendre au fil de celles-ci?)



2.1. Cycles de développement d'un système d'information

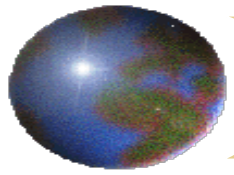
- ✿ **Le cycle de vie:** Présente la manière de conduire le projet. Il se situe sur une échelle de temps qui nous mène du point de départ à l'exploitation du système, en passant par sa création, sa maturité et sa maintenance.
- ✿ **Le cycle d'abstraction:** C'est le découpage en ensembles homogènes de préoccupations :
 - ✦ le niveau conceptuel qui détermine les choix de gestion
 - ✦ le niveau organisationnel qui détermine les choix d'organisation
 - ✦ le niveau technique qui détermine les contraintes techniques
- ✿ **Le cycle de décision:** Il représente l'ensemble des choix qui doivent être fait durant le déroulement du cycle de vie





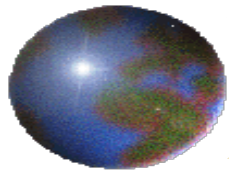
2.2.

Le cycle de vie d'un logiciel



2.2. Le cycle de vie d'un logiciel

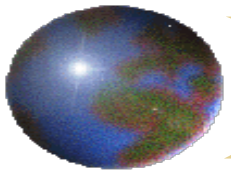
- ✚ Le « **cycle de vie d'un logiciel** » (*en anglais software **lifecycle***), désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition.
- ✚ L'**objectif** d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la vérification du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.
- ✚ L'**origine de ce découpage** provient du constat que les erreurs ont un coût d'autant plus élevé qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs au plus tôt et ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés.



2.2. *Le cycle de vie d'un logiciel*

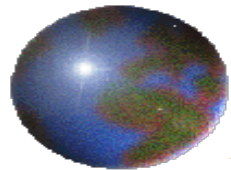
De façon générale, le cycle de vie du logiciel comprend les activités suivantes :

- ✚ **Définition des objectifs**, consistant à définir la finalité du projet et son inscription dans une stratégie globale.
- ✚ **Analyse des besoins et faisabilité**, c'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes.
- ✚ **Conception générale**. Il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel.
- ✚ **Conception détaillée**, consistant à définir précisément chaque sous-ensemble du logiciel.
- ✚ **Codage (Implémentation ou programmation)**, soit la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.
- ✚ **Tests unitaires**, permettant de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux spécifications.



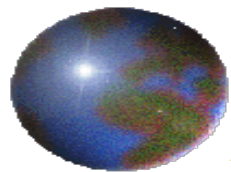
2.2. *Le cycle de vie d'un logiciel*

- ✚ **Intégration**, dont l'objectif est de s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Elle fait l'objet de tests d'intégration consignés dans un document.
- ✚ **Qualification (ou recette)**, c'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales.
- ✚ **Documentation**, visant à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.
- ✚ **Mise en production**,
- ✚ **Maintenance**, comprenant toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.



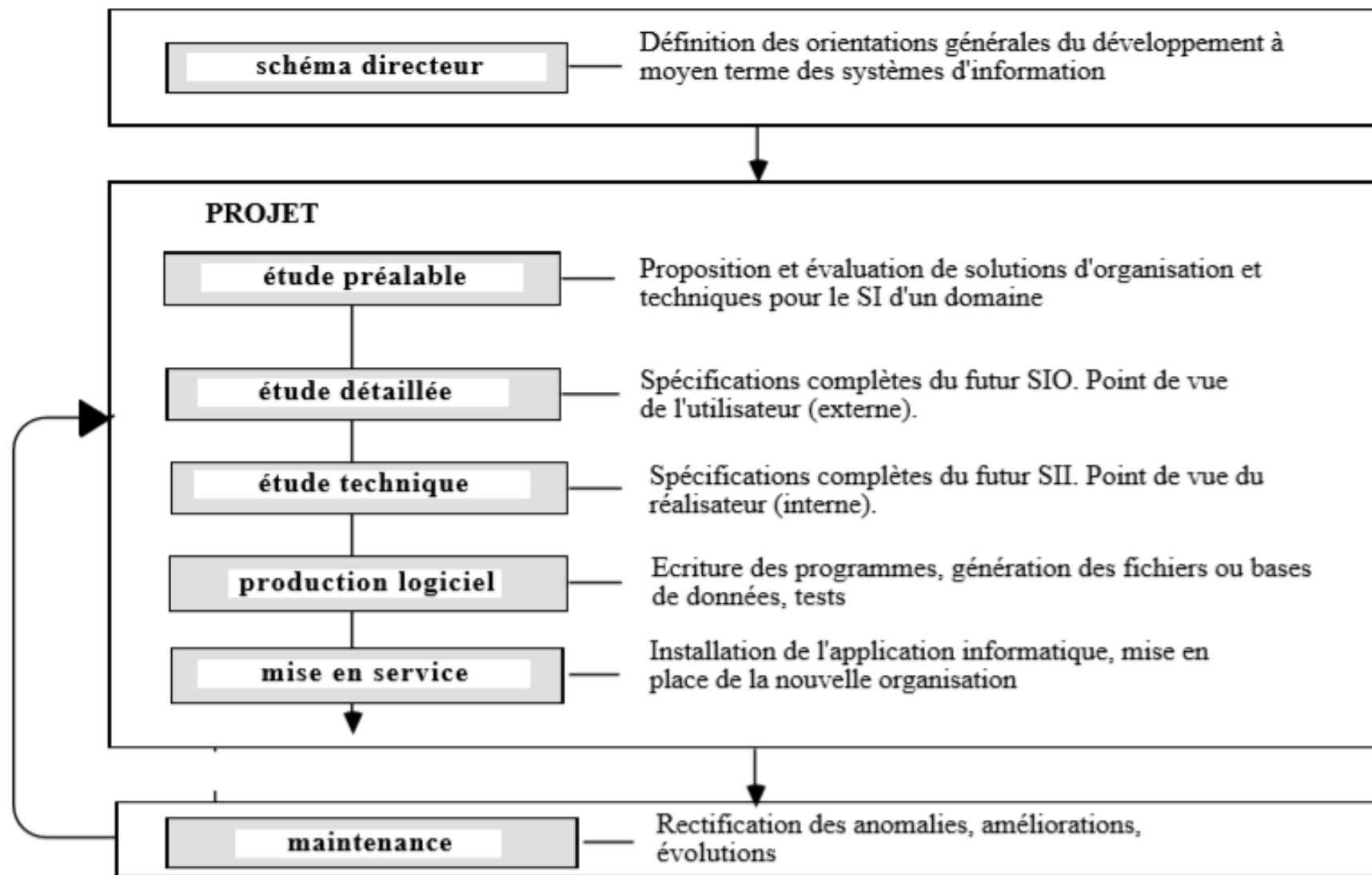
2.2. *Le cycle de vie d'un logiciel*

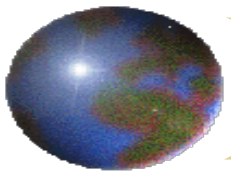
- ✚ La séquence et la présence de chacune de ces activités dans le cycle de vie dépend du choix d'un modèle de cycle de vie (voir section suivante) entre le client et l'équipe de développement.
- ✚ Plus classiquement, le cycle de vie se résume en une démarche comprenant les étapes successives suivantes :
 1. **Analyse / Conception** (Schéma directeur, Etude préalable, Etude détaillée)
 2. **La réalisation** (Etude Technique, Production Logicielle, Mise en service)
 3. **La Maintenance**



2.2. Le cycle de vie d'un logiciel

1. le schéma directeur
2. l'étude préalable
3. l'étude détaillée
4. l'étude technique
5. La production
6. la qualification
7. la maintenance





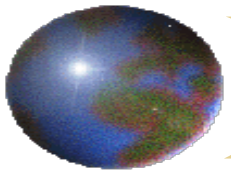
2.2. *Le cycle de vie d'un logiciel*

1. *Le schéma directeur*

- ✚ Etude globale du SI : Découpage en domaines (ex: domaine « Approvisionnement »).

- ✚ **Buts :**
 - ✚ Définir les grandes orientations politiques et stratégiques de l'entreprise
 - ✚ Définir les besoins en SI en fonction de la stratégie de l'entreprise
 - ✚ Fixer les cadres budgétaires, la stratégie des besoins en personnel et les contraintes diverses liées à l'environnement
 - ✚ Fixer les lignes directrices des développements informatiques
 - ✚ Définir les projets nécessaires à l'élaboration ou l'évolution du SI

- ✚ **Documents produits :**
 - ✚ Le schéma directeur
 - ✚ Le plan de développement informatique.



2.2. *Le cycle de vie d'un logiciel*

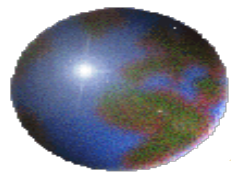
1. *Le schéma directeur*

✚ Résultats attendus:

- ✚ définition précise des domaines,
- ✚ une planification du développement de chaque domaine et un plan détaillé, année par année, des applications qui doivent être réalisées.

✚ Le schéma directeur peut se faire en **plusieurs étapes.**

- ✚ Etape 1 : définir les objectifs et les structures de travail.
- ✚ Etape2 : déterminer les composants de l'entreprise.
- ✚ Etape 3 : formaliser et chiffrer les solutions.
- ✚ Etape 4 : planifier les actions.
- ✚ Etape 5 : définir la structure de pilotage.

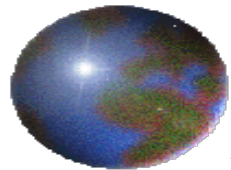


2.2. *Le cycle de vie d'un logiciel*

2. *L'étude préalable*

Une étude préalable (étude de faisabilité) s'applique à un domaine ou à une grande fonction de l'entreprise. Résultats attendus:

- ✚ Objectif: obtenir le descriptif complet de la nouvelle solution pour le domaine envisagé en plusieurs phases, de sorte à envisager les différentes hypothèses possibles et à s'orienter progressivement vers la solution optimum.
- ✚ Point final de l'étude préalable est de:
 - ✚ décider d'une solution type en parfaite connaissance de cause quant à sa faisabilité: coût, rentabilité, délai, budget, moyens à mettre en œuvre, impact organisationnel, ...
 - ✚ établir le cahier des charges pour la réalisation; sélectionner un sous-traitant ou un progiciel.



2.2. Le cycle de vie d'un logiciel

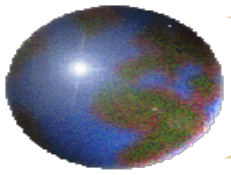
2. L'étude préalable

- ✦ Pour mener à bien une étude préalable, on part de la situation existante. L'étude préalable est faite par des utilisateurs et des organisateurs en liaison avec la direction générale et les informaticiens.

On en distingue plusieurs étapes dont chacune est divisée en phases:

✦ Étape 1 : Étude de l'existant

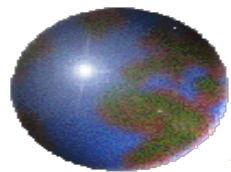
- ✦ **Phase 1** : Rédiger une Fiche de présentation générale du problème. Cette fiche situe le domaine dans le système d'information, à partir du schéma directeur et de la demande de la direction générale. Par interview de la direction générale, se faire préciser les grands objectifs relatifs au domaine et les consigner dans une Fiche d'objectifs D.G.
- ✦ **Phase 2** : Procéder à des interviews des postes de travail concernés et rédiger des comptes rendus d'interviews, en y joignant tous les documents concernés actuellement en vigueur. S'il y a des traitement qui se font à l'aide ou par ordinateur, joindre les fiches de description de fichiers actuels.



2.2. *Le cycle de vie d'un logiciel*

2. *L'étude préalable*

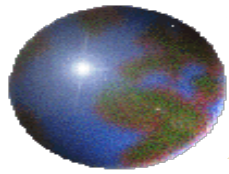
- ❖ **Phase 3** : A partir des comptes rendus, établir les documents suivants:
 - le graphe de circulation actuel;
 - la carte de circulation des informations du domaine ;
 - le graphe des flux ;
 - le MOT actuel ;
- ❖ **Phase 4** : à partir des documents recueillis lors des interviews, établir un dictionnaire des données actuel pour toutes les données manipulées et épurer ce dictionnaire.
- ❖ **Phase 5** : a partir du dictionnaire des données, construire la **structure d'accès théoriques (SAT)** en ne prenant que les données élémentaires, sans les données calculées, en listant toutes les dépendances fonctionnelles et en éliminant les transitivités. En déduire **une représentation graphique du MCD actuel**.
- ❖ **Phase 6** : Déduire les règles de gestion à partir du MCD et du MCT actuels ;



2.2. *Le cycle de vie d'un logiciel*

2. *L'étude préalable*

- ❖ **Phase 7** : à partir des comptes rendus d'interviews établir une **fiche de synthèse** faisant état des points les plus épineux, des souhaits des utilisateurs, en s'efforçant de critiquer l'organisation actuelle. Dégager de cette fiche un certain nombre d'objectifs de la "fiche d'objectifs D.G.", établir une liste hiérarchisée de tous les objectifs sur une fiche d'objectifs synthèse. Faire approuver celle-ci par la direction générale et par les divers responsables.
- ❖ **Phase 8** : étudier les interfaces avec les autres domaines (d'après la carte générale de circulation de l'information du schéma directeur, par exemple) et les consigner dans une fiche des interfaces.



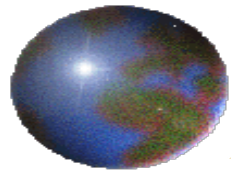
2.2. Le cycle de vie d'un logiciel

✚ Étape 2 : Scénarios futurs

- ✚ **Phase 1** : à partir de la fiche d'objectif synthèse et des règles de gestion actuelles, inventorier les règles de gestion futures et les inscrire sur une fiche de règles de gestion futures. A partir du MCD actuel, de la fiche des interfaces et des règles de gestion futures, établir le **MCD futur**.
- ✚ **Phase 2** : à partir du MCT actuel et des règles de gestion futures, concevoir le **MCT futur**.
- ✚ **Phase 3** : pour chaque scénario faire l'étude organisationnelle qui donnera :
 - une fiche des règles d'organisation ;
 - une esquisse de MOT futur.

Faire ensuite l'étude opérationnelle (toujours par scénario) en définissant le matériel et les logiciels nécessaires. S'il existe plusieurs sites, cette étude tient compte des aspects spécifiques de chaque site. L'étude opérationnelle donnera lieu à :

- une fiche de configuration de matériel pour l'architecture générale et pour chaque site s'il y en a plusieurs ;
- une liste des logiciels.



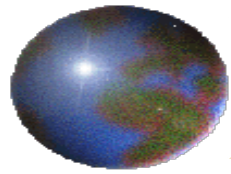
2.2. *Le cycle de vie d'un logiciel*

✚ **Étape 3 : Une évaluation des Scénarios**

Pour chaque scénario rédiger un rapport d'évaluation du scénario comportant :

▣ **une évaluation des coûts de :**

- matériel (achat, location)
- stockage
- logiciels (programmation, achat de progiciels...)
- traitement (unités centrales, consommation d'électricité, ...)
- communication (location de lignes de transmission)
- exploitation (fournitures de consommable, entretien des locaux ...)
- maintenance (contrats de maintenance chez le constructeur)
- personnel (utilisateurs, service informatique, sous-traitant, ...)
- formation (utilisateurs, informaticiens)



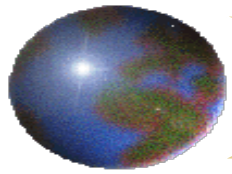
2.2. *Le cycle de vie d'un logiciel*

❏ **une évaluation des avantages :**

- quantifiables et chiffrables au plan financier (recette attendues, économies de personnel ...)
- quantifiables et non chiffrables au plan financier (diminution des temps de gestion, simplification des procédures, ...)
- non quantifiables (rapidité des traitements, aide à la décision, image de marque, fidélisation d'une clientèle, ...)

❏ **une évaluation de l'impact sur l'organisation:**

- impact sur les postes de travail
- acceptabilité du personnel



2.2. *Le cycle de vie d'un logiciel*

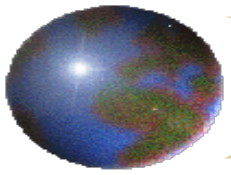
❏ **une évaluation de la faisabilité**

- en matériels (rédiger un compte rendu de l'étude technique)
- en logiciels
- en personnel (personnel disponible, embauche, formation, sous-traitance)

❏ **une évaluation approximative des délais**

- de livraison des matériels
- de programmation des logiciels (délais rarement respectés)
- de recrutement et de formation - une évaluation de la mise en œuvre
- cadencement du lancement des applications
- périodes transitoires
- etc.

L'étude préalable donne lieu à un document appelé **dossier de choix**.



2.2. Le cycle de vie d'un logiciel

3. L'étude détaillée

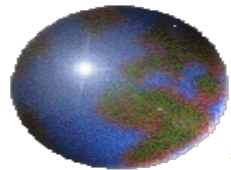
- ✚ L'étude préalable ne porte que sur les processus majeurs. La description des données et des traitements y est succincte.
- ✚ L'étude détaillée va décrire tous les processus composants le fonctionnement du futur système; définir précisément les informations utilisées et mémorisées; spécifier complètement les tâches à effectuer.

Elle se déroule en plusieurs étapes:

✚ **Étape 1 : Étude générale du MOT futur**

Cette étude se fait en suivant les phases suivantes :

- réalisation du tableau des procédures fonctionnelles ;
- construction du diagramme d'enchaînement des procédures fonctionnelles ;
- construction du graphe de circulation.



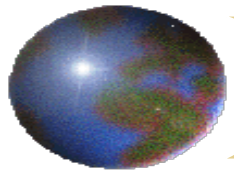
2.2. Le cycle de vie d'un logiciel

3. L'étude détaillée

✦ Étape 2 : Étude poussée de chaque procédure fonctionnelle

dans cette étape, chaque procédure fonctionnelle est étudiée de manière poussée. Cela donne lieu à :

- une fiche de description de la procédure fonctionnelle
- une description des documents éventuelles
- des tables de décision éventuelles
- une description éventuelle des états de sortie
- pour les transactions :
 - * des grilles d'écrans (à faire approuver par les utilisateurs concernés)
 - * des grilles de contrôles
 - * une fiche de répartition des tâches entre l'homme et la machine (à faire approuver par les utilisateurs)
- le modèle externe non validé



2.2. *Le cycle de vie d'un logiciel*

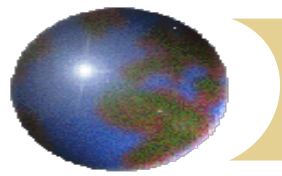
3. *L'étude détaillée*

✦ Étape 3 : Validation du MCD

Les données et les traitements ayant été étudiés de manière indépendante, la validation du MCD permet de confronter les données aux traitements et de rendre l'ensemble cohérent.

✦ Étape 4 : Passage au MLD

L'étude détaillée donne lieu, pour chaque application à un cahier des charges utilisateurs.



2.2. *Le cycle de vie d'un logiciel*

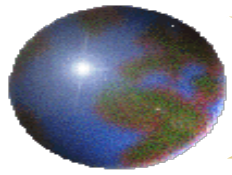
4. *L'étude technique*

L'étude technique est la traduction informatique des spécifications issues de l'étude détaillée. Elle s'occupe des spécifications techniques pour la réalisation.

Les objectifs de l'étude technique sont de présenter les spécifications informatiques nécessaires à la production du logiciel.

A cet effet, elle définit complètement :

- ❑ la structure de mémorisation informatique des données (fichiers ou base de données) ;
- ❑ l'architecture interne du système
- ❑ les procédures de sécurités
- ❑ le plan de réalisation



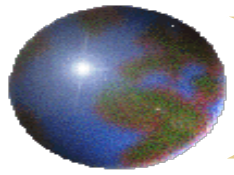
2.2. *Le cycle de vie d'un logiciel*

4. *L'étude technique*

L'étude technique peut être menée en suivant les étapes suivantes :

✚ **Étape 1 : Structuration des données**

- ☐ établir la description physique des données à partir du MLD (Modèle Logique des données)
- ☐ définir les clés d'accès
- ☐ établir une quantification de l'activité des éléments de stockage des données
- ☐ procéder à l'optimisation physique
- ☐ déduire l'allocation des espaces physiques, l'organisation et le mode d'accès
- ☐ décrire les dispositifs de protection d'accès et de confidentialité



2.2. Le cycle de vie d'un logiciel

4. L'étude technique

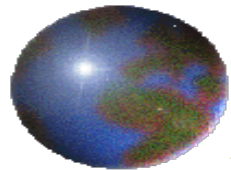
✦ Étape 2 : spécification de l'architecture interne du système

découpage des Unité Fonctionnelle (UF) en Unités de Traitement (UT)

- ☐ découper chaque UF en UT

découpage des UT en modules

- ☐ découper chaque UT en modules
- ☐ établir la liste des Entrées/Sorties pour chaque module
- ☐ établir la liste des données utilisées ou échangées
- ☐ établir une description pour chaque module sous forme d'algorithmes ou d'organigramme
- ☐ déterminer les jeux d'essai à élaborer répondant aux exigences de la qualité



2.2. *Le cycle de vie d'un logiciel*

4. *L'étude technique*

terminer de concevoir les Entrées/Sorties

- ☐ définir les spécifications des écrans ainsi que les dialogues associés
- ☐ définir les formats détaillés et les spécifications des sorties
- ☐ établir une liste des entrées/sorties pour chaque UT

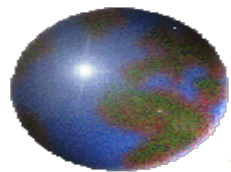
✦ **Étape 3 : les procédures techniques de sécurité**

contrôle des données :

- ☐ description des caractéristiques liées aux contrôles des données

sécurité et confidentialité :

- ☐ description des mécanismes relatifs à la sécurité et à la confidentialité des données
- ☐ description les procédures de reconstitution des supports de données, de redémarrage et de sauvegarde



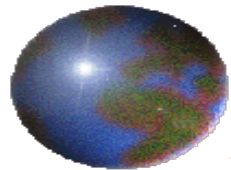
2.2. *Le cycle de vie d'un logiciel*

4. *L'étude technique*

✦ Étape 4 : planification de la réalisation

- ☐ réaliser le planning pour les phases suivantes

L'étude technique donne lieu à un **cahier des charges technique**



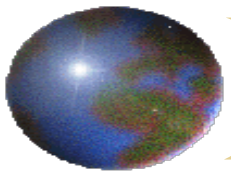
2.2. *Le cycle de vie d'un logiciel*

5. La production

La production consiste en l'écriture effective des programmes dans le langage approprié, à la génération des fichiers ou des bases de données et aux tests. La production du logiciel doit permettre de faciliter la maintenance future.

Ses objectifs sont :

- ☐ la production du logiciel conforme aux spécifications de l'étude technique
- ☐ de tester techniquement le logiciel de manière qu'il soit prêt à être livré aux utilisateurs



2.2. *Le cycle de vie d'un logiciel*

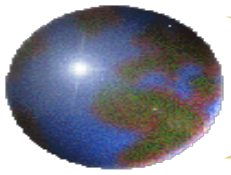
5. *La production*

La production du logiciel peut se faire en suivant les étapes suivantes :

✚ **Étape 1 : Programmation de l'application**

durant laquelle il faudra :

- ☐ intégrer les normes et standards de programmation
- ☐ organiser l'environnement de programmation
- ☐ définir une organisation pour permettre la gestion de l'environnement de programmation (matériel, temps machine, ..)



2.2. *Le cycle de vie d'un logiciel*

5. *La production*

coder les modules

- ☐ transformer les spécifications techniques des modules en code source
- ☐ vérifier et mettre au point le code source
- ☐ corriger les défauts et erreurs constatées

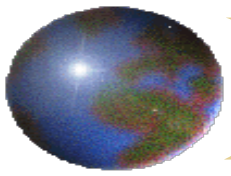
préparer les jeux d'essai

- ☐ définir les jeux d'essai
- ☐ préparer les données en vue des tests unitaires et d'enchaînement de modules

✦ **Étape 2 : tests, mise au point**

tests unitaires

- ☐ chaque module écrit, correspondant à une Unité de Traitement, doit faire l'objet d'un premier test à partir du jeu d'essai



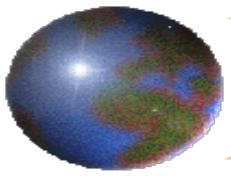
2.2. *Le cycle de vie d'un logiciel*

5. *La production*

tests d'intégration

- ces tests doivent permettre de vérifier le bon fonctionnement de l'ensemble du logiciel. Ils sont généralement effectués par le responsable technique du projet.

La production donne lieu à un **dossier de réalisation**.



2.2. *Le cycle de vie d'un logiciel*

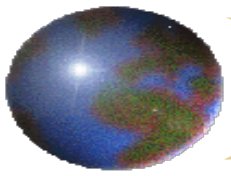
6. *La mise en service*

- ✦ L'objectif principal de la mise en service est de rendre opérationnel le système d'information. Elle consiste à installer l'ensemble du logiciel développé et à mettre l'ensemble du SI au service des utilisateurs. Elle permet de conclure l'ensemble du cycle conception développement.
- ✦ Son objectif principal est de mettre en place tous les moyens nécessaires à la réception et au lancement du nouveau SI.

La mise en service se déroule en deux étapes:

- ✦ **Étape 1 : l'installation**

elle a pour but de livrer le logiciel aux utilisateurs et de préparer le démarrage de l'exploitation. Elle comprend :



2.2. *Le cycle de vie d'un logiciel*

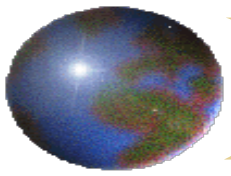
6. *La mise en service*

- ❖ la commande, l'installation, le contrôle et la réception du matériel ;
- ❖ la mise en place des procédures nouvelles liées aux changements d'organisation ;
- ❖ l'organisation et le déroulement de la formation des utilisateurs et des exploitants ;
- ❖ l'exécution des jeux d'essai utilisateurs ;
- ❖ l'appréciation de la conformité des résultats ;
- ❖ la réception provisoire.

✚ **Étape 2 : la mise en exploitation**

elle consiste en :

- ❖ l'exploitation en grandeur réelle du nouveau système
- ❖ le lancement progressif du nouveau système en parallèle au système existant si possible

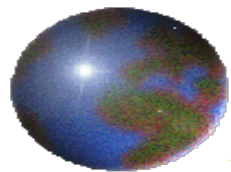


2.2. *Le cycle de vie d'un logiciel*

6. *La mise en service*

- ❏ le fonctionnement en vrai grandeur pendant une période déterminée
- ❏ le bilan de l'exploitation
- ❏ la réception définitive du produit

Elle donne lieu au dossier d'exploitation et au **manuel utilisateur**.



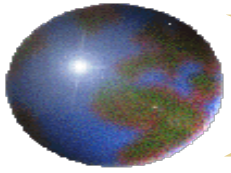
2.2. *Le cycle de vie d'un logiciel*

7. *La maintenance*

- ✿ La maintenance est définie selon BOEHM comme l'ensemble des opérations de modification d'un logiciel opérationnel laissant intactes ses fonctionnalités de base. Elle consiste en la rectification des anomalies, les améliorations et les évolutions.

Ses objectifs sont :

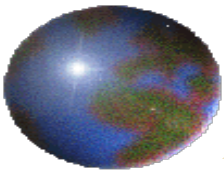
- ✿ maintenir le système en fonctionnement ;
- ✿ rectifier les anomalies de fonctionnement et prendre en compte les demandes d'évolution ;
- ✿ établir un scénario de développement des versions ;
- ✿ planifier les corrections apportées au système pour minimiser les coûts d'intervention ;
- ✿ s'assurer du bon fonctionnement des nouvelles révisions avant toute nouvelle mise en exploitation ;



2.2. *Le cycle de vie d'un logiciel*

7. La maintenance

- ❏ procéder à cette mise en exploitation après l'accord des parties prenantes ;
- ❏ assurer une bonne gestion des différentes configurations du logiciel par la tenue à jour de la documentation du système.



2.3.

Les différents modèles de cycles de vie

Modèle en Cascade
ou Waterfall

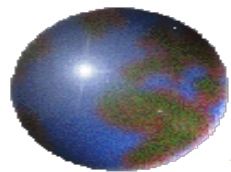
Modèle en V

Modèle en itératif
et incrémental

Modèle en Spirale

Modèle RAD

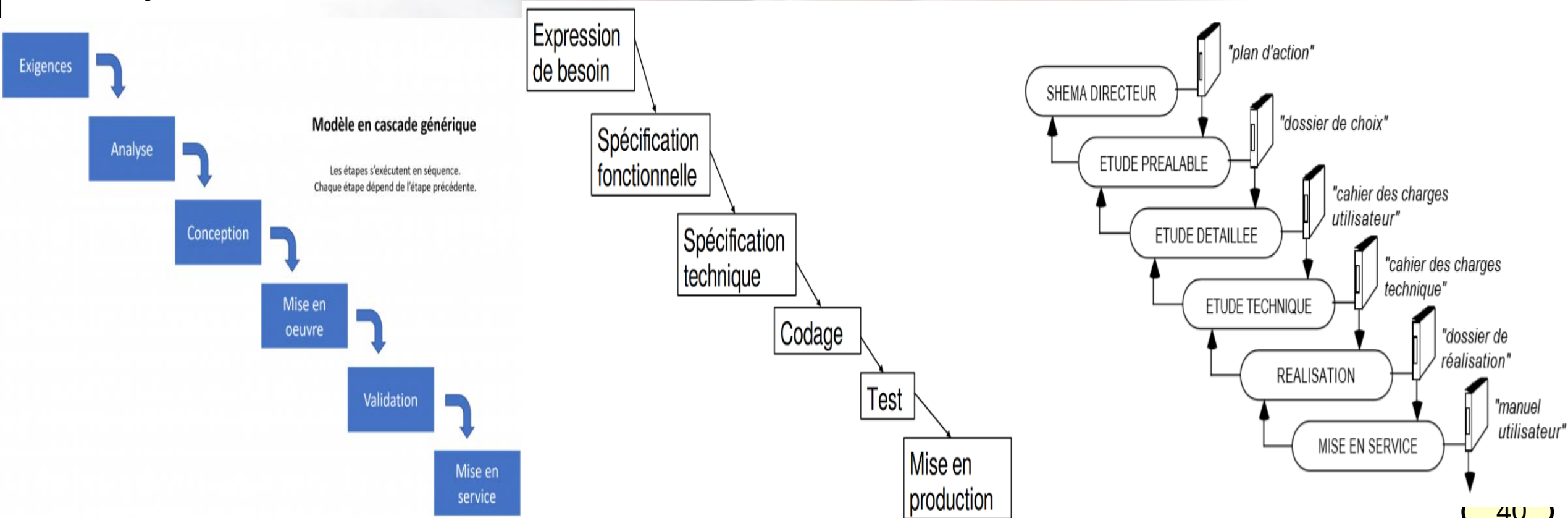
Méthodes agiles

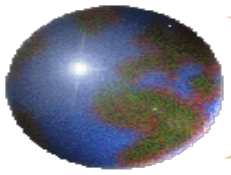


2.3. Les différents modèles de cycles de vie

1. Cycle de vie en cascade (ou Waterfall en anglais)

- ✚ Dans le cycle de vie en cascade, les étapes se succèdent dans le temps. Chaque étape produit des documents qui sont utilisés pour en vérifier la conformité avant de passer à la suivante.
- ✚ Auteur: Herbert D. Benington en 1956
- ✚ Le cycle de vie en cascade est souvent schématisé de la manière suivante :





2.3. Les différents modèles de cycles de vie

1. Cycle de vie en cascade (ou Waterfall en anglais)

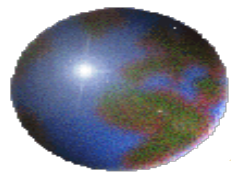
✚ Avantages:

- sa simplicité et sa souplesse
- offrent une vision très précise des délais de réalisation des projets

✚ Inconvénients:

- Dès qu'une modification nécessitant la reprise d'une étape, c'est tout le processus qui est impacté
- chaque étape ne validant que l'étape précédente, on prend un risque de voir le projet s'écarter petit à petit de son objectif initial, juste un peu à chaque étape, pour aboutir à un résultat final trop éloigné pour être satisfaisant.

Ce risque est bien réel. C'est la raison pour laquelle a été créé le Cycle en V, avec son système élaboré de validations croisées. Sauf que ces validations prennent un temps conséquent, et on perd alors encore plus de souplesse.



2.3. Les différents modèles de cycles de vie

2. Cycle de vie en V

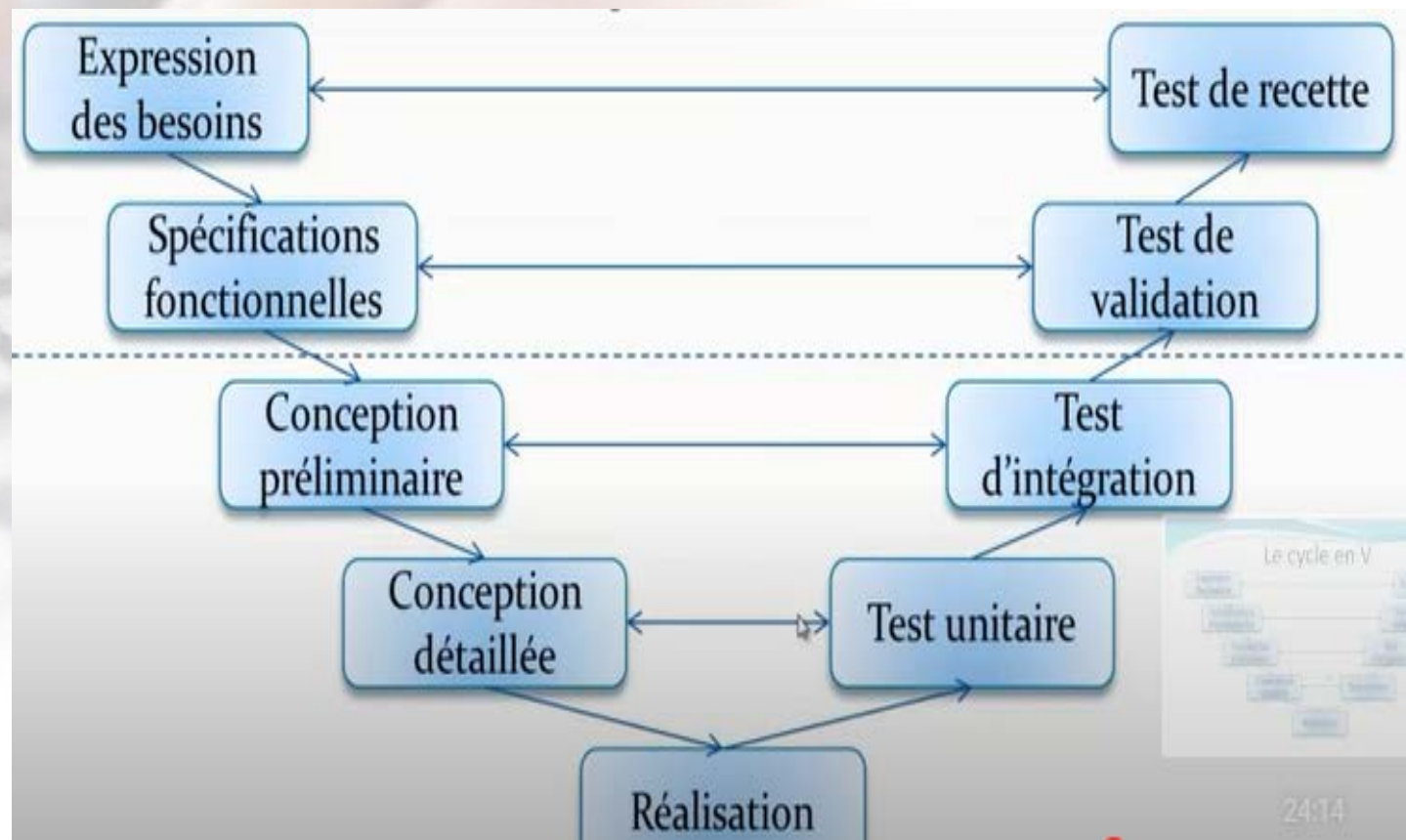
Le modèle de cycle de vie en V est un modèle de gestion de projet qui part du principe que les procédures de vérification de la conformité du logiciel aux spécifications doivent être élaborées dès les phases de conception.

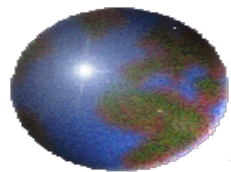
Il découle du modèle en cascade.

Il a été Développé dans les années 1980.

❑ Le modèle en V met l'accent sur les tests: un plan de test est préparé avant l'implémentation pour chaque niveau.

❑ Ce modèle exclu les allées et venues entre le cahier de charge et la conception du SI.





2.3. Les différents modèles de cycles de vie

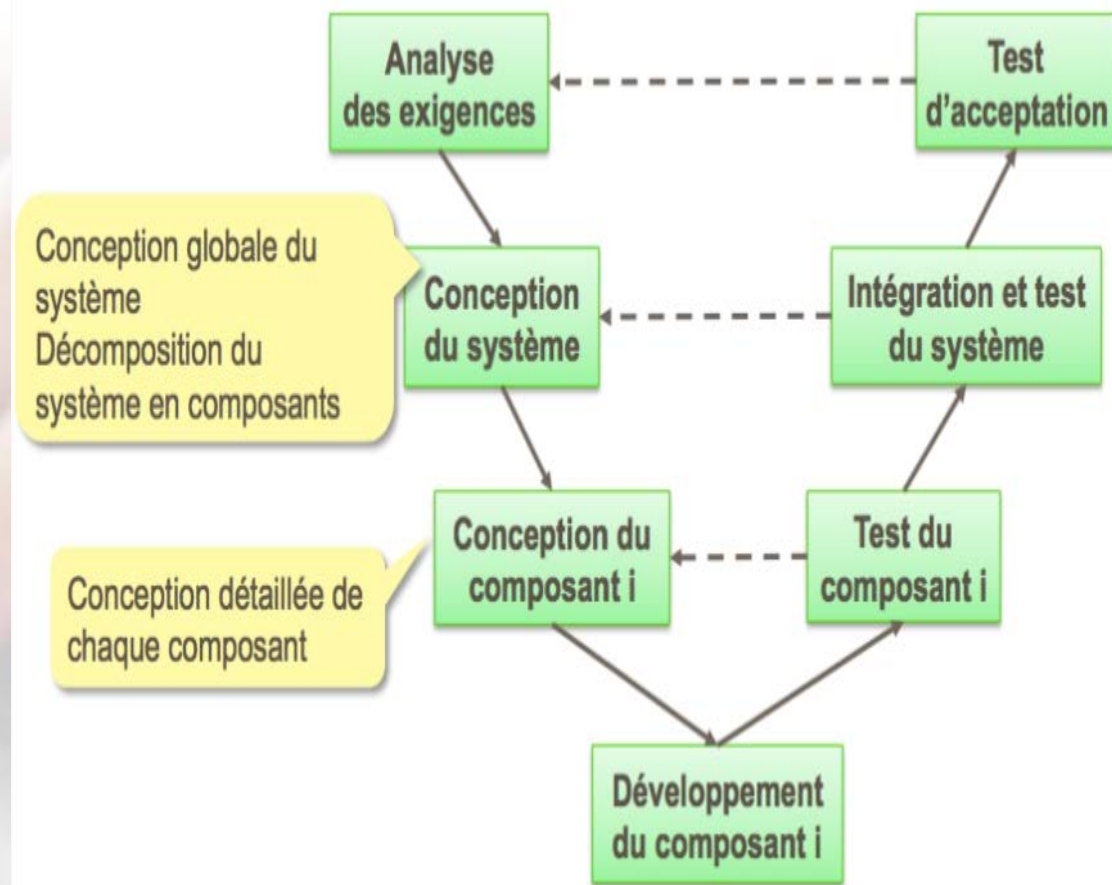
2. Cycle de vie en V

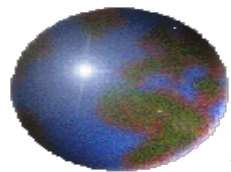
✚ Avantages:

- évite de revenir en arrière incessamment pour redéfinir les spécifications initiales
- facile à mettre en œuvre

✚ Inconvénients:

- L'inconvénient principal du cycle en V se résume en deux mots : l'effet tunnel. Le cycle en V supporte mal les changements, ce qui est à la fois sa force et sa principale faiblesse. Après l'expression de besoin, le client ne voit pas ce qui est fait avant la livraison en recette. Il peut donc rester plusieurs mois sans "voir" le résultat. Il n'a que les comités de pilotage pour l'informer de l'avancement.





2.3. Les différents modèles de cycles de vie

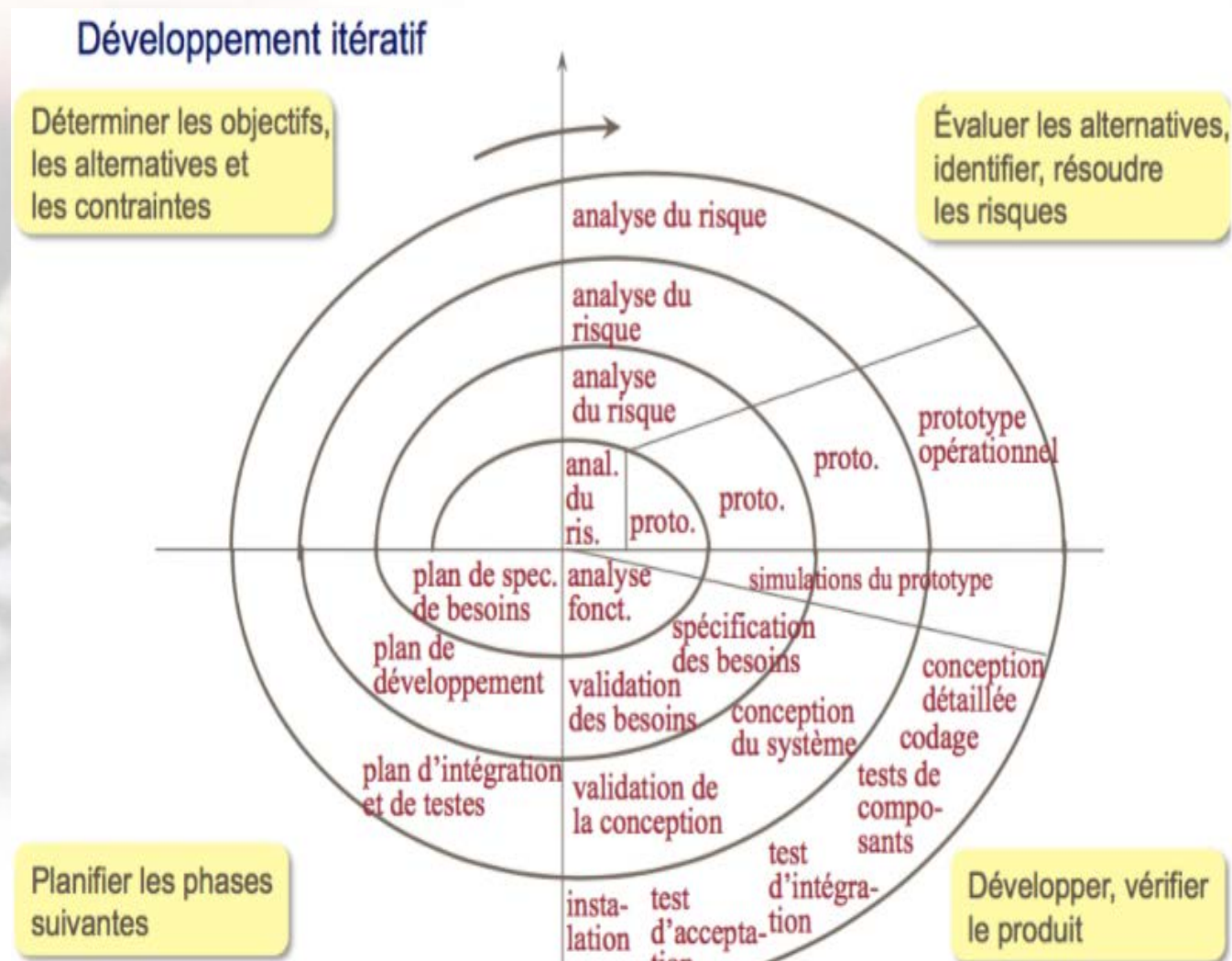
3. Cycle de vie en spirale

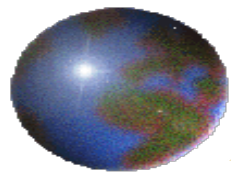
Le cycle de vie en spirale est un modèle générique de cycle de vie évolutif qui a été proposé par Barry W. Boehm en 1984.

Ce modèle, axé sur la maîtrise et la réduction des risques, est davantage un cadre de travail guidant la construction d'une démarche spécifique de projet, plutôt qu'une démarche formalisée.

Chaque boucle de spirale permet :

- d'identifier les objectifs propres de la boucle
- les moyens alternatifs pour atteindre les objectifs
- les contraintes de chaque alternative



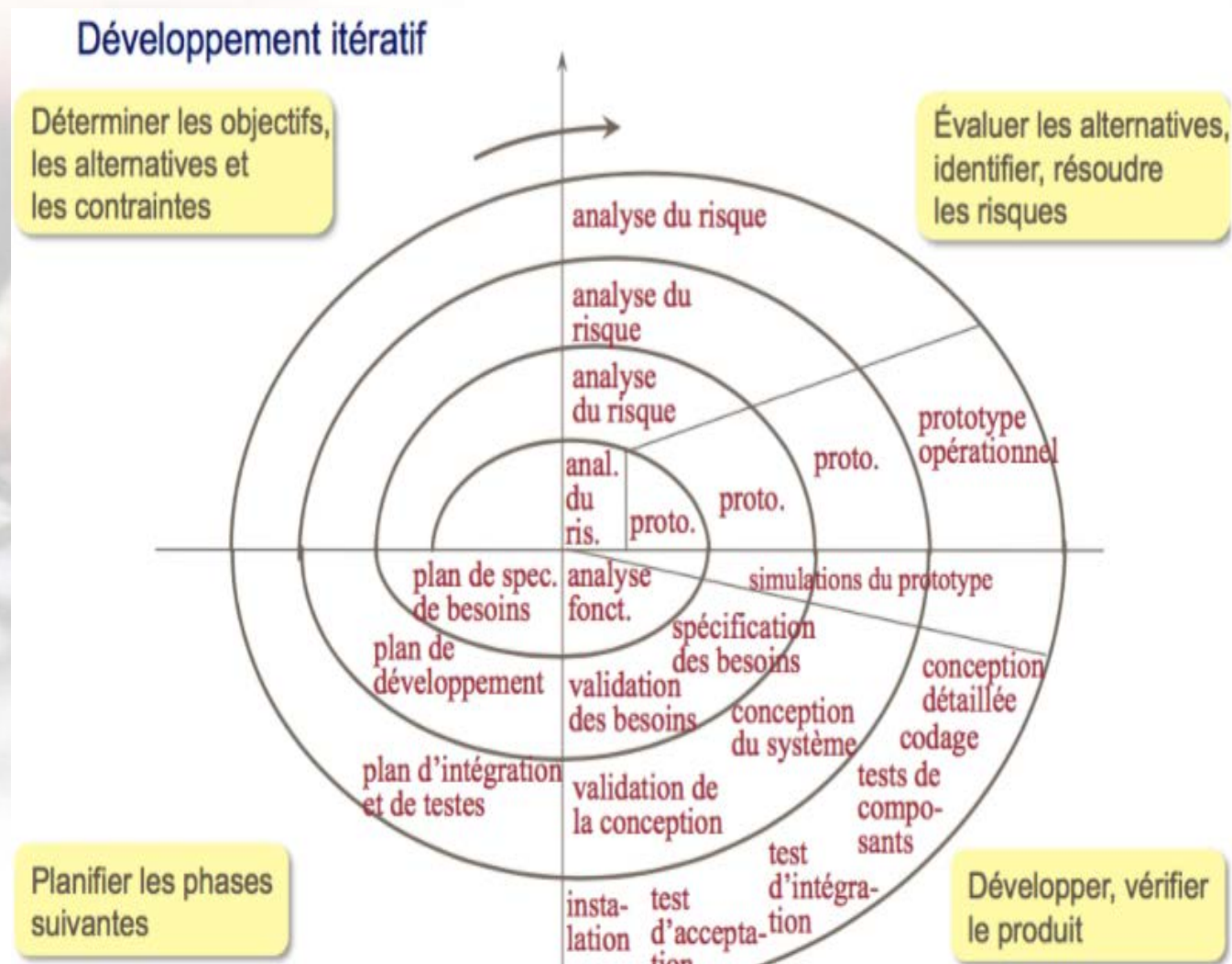


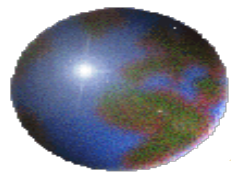
2.3. Les différents modèles de cycles de vie

3. Cycle de vie en spirale

On distingue quatre phases dans le déroulement du cycle en spirale :

- détermination des objectifs, des alternatives et des contraintes ;
- analyse des risques, évaluation des alternatives ;
- développement et vérification de la solution retenue ;
- revue des résultats et vérification du cycle suivant.





2.3. Les différents modèles de cycles de vie

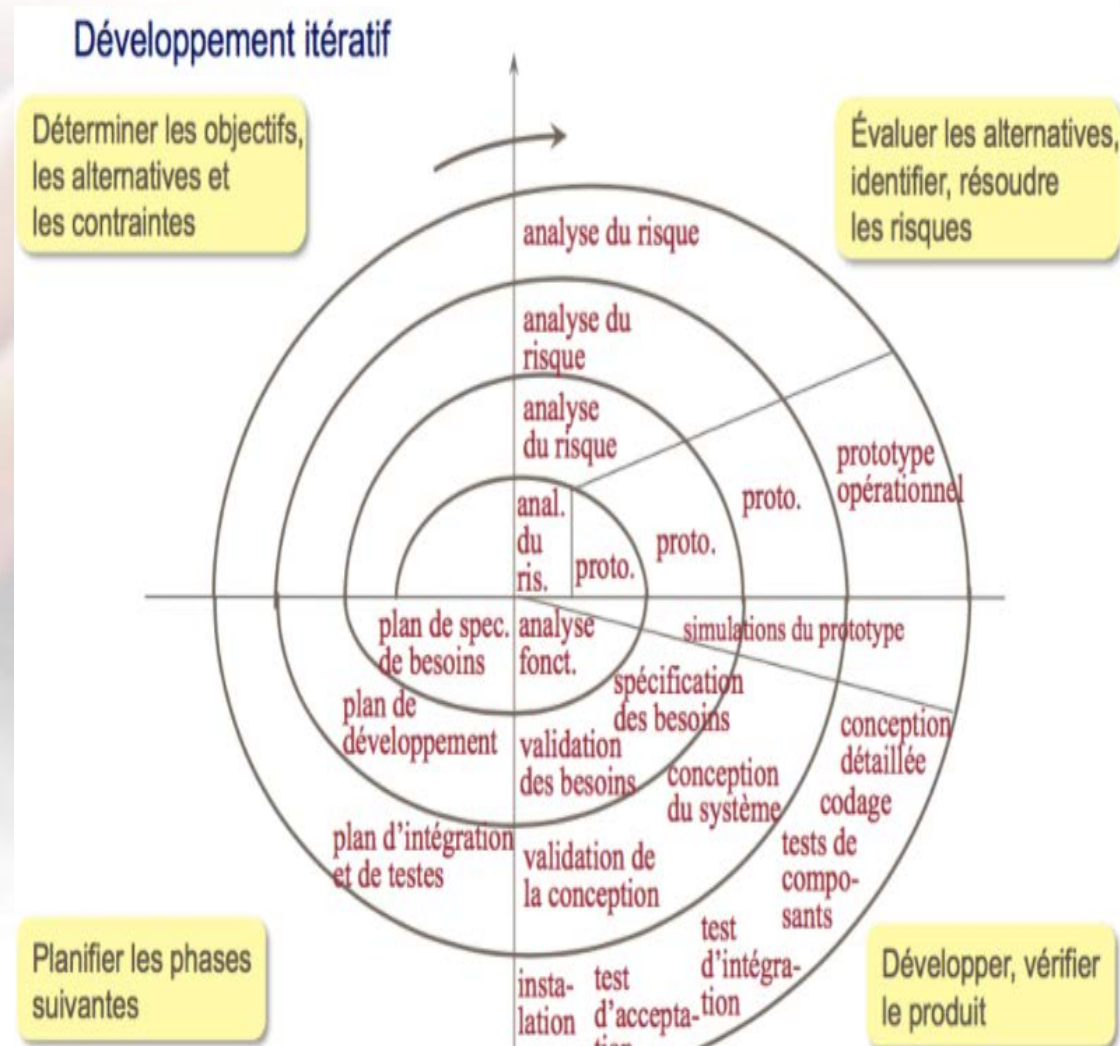
3. Cycle de vie en spirale

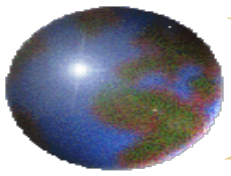
✚ Avantages:

- Le client est plus impliqué dans la vie du projet que dans les modèles en V et en cascade.
- Ce modèle favorise plus les interactions entre les équipes du projet

✚ Inconvénients:

- L'inconvénient principal du cycle en V se résume en deux mots : l'effet tunnel. Le cycle en V supporte mal les changements, ce qui est à la fois sa force et sa principale faiblesse. Après l'expression de besoin, le client ne voit pas ce qui est fait avant la livraison en recette. Il peut donc rester plusieurs mois sans





2.3. Les différents modèles de cycles de vie

4. Cycle de vie itératif et incrémental

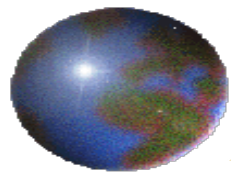
Il permet de construire progressivement le système souhaité.

Il répète les étapes du cycle en cascade ou en V jusqu'à l'obtention du produit souhaité.

A chaque itération, il y'a une réalisation d'un prototype qui corrige ou complète son précédent. (comme dans le cycle spiral)

L'utilisation d'un prototype constitue une base de travail concrète et mesurable dans le cycle itératif et incrémental.





2.3. Les différents modèles de cycles de vie

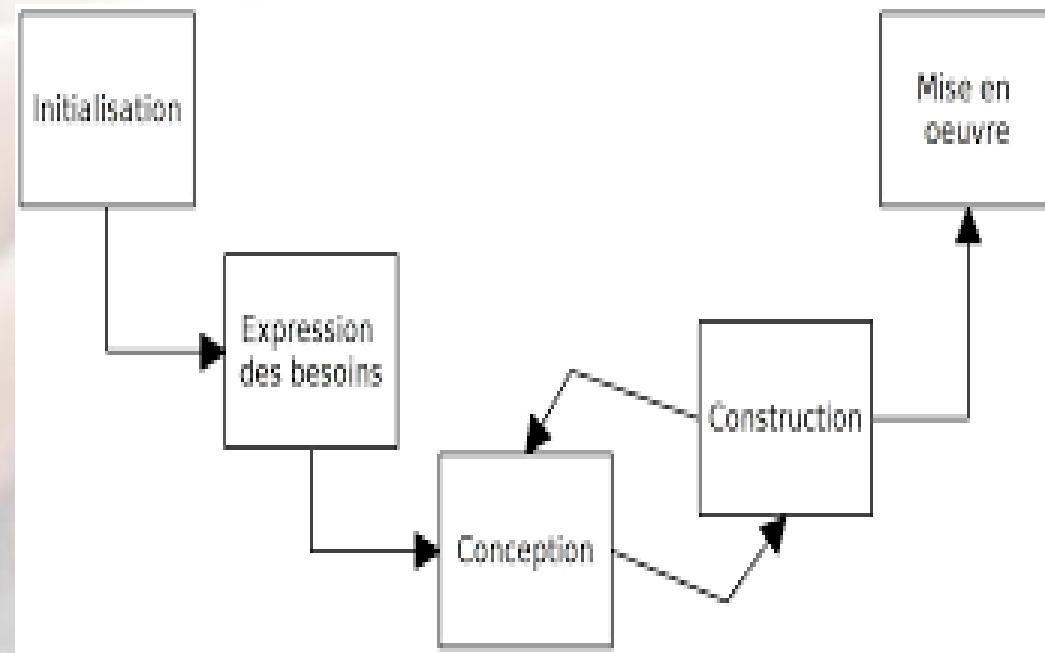
5. Cycle de vie en RAD

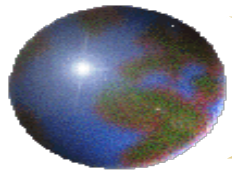
RAD (Rapid Application Développement) est un modèle qui conjugue le modèle en cascade et le modèle itératif. Son but est de livrer rapidement un minimum de fonctions viables pour assurer un retour sur investissement rapide et éviter l'effet tunnel.

Il se base sur l'utilisation de prototype, une démarche participative de tout les intervenants du projet, une priorisation des délais

Il se caractérise par:

- Un cycle de développement accéléré
- Des validations fréquentes
- Une approche par composant.





2.3. *Les différents modèles de cycles de vie*

6. *Les méthodes agiles*

Les méthodes agiles ont bousculé les modes de gestion classique de cycle de vie de logiciel. Elles tirent leur racines de la méthode RAD.

Les méthodes agiles, apparues dans les années 2000, prennent en compte dans leur modèle de cycle de vie 3 exigences:

- Une forte participation entre développeurs et utilisateurs
- Des livraisons fréquentes de logiciel
- Et une prise en compte de possibles changements dans les besoins des utilisateurs au cours du projet.

Exemples de méthodes agiles: **SCRUM, XP, DSDM** etc.