

Analyse et spécification des besoins

I- L'analyse des besoins :

La définition de l'analyse des besoins constitue en principe la première phase du cycle de vie d'un logiciel et consiste en l'activité visant à établir quelle fonctionnalité le système doit fournir et les contraintes auxquelles il sera soumis.

Cette étape doit aboutir à l'élaboration des documents de définition des besoins (le cahier des charges, souvent exprimés en langage naturel).

Les besoins exprimés dans ce document doivent constituer un ensemble complet et cohérent, la complétude signifie que tous les services requis par l'utilisateur doivent être mentionnés et la cohérence que les définitions des besoins ne doivent pas être contradictoire.

1. classification des besoins :

Les besoins peuvent être classifiés ainsi:

- Besoins fonctionnels: correspondant aux services attendues par l'utilisateur.
- Besoins en matière de BD: dans une BD le système y trouve ces informations et y ajoute d'autres pendant son fonctionnement. Ces informations doivent refléter l'activité des utilisateurs d'une manière intégrée et cohérente.
- Besoins non fonctionnels: correspondant aux restrictions (limitations) ou aux contraintes accès sur des fonctions du système tel que: le temps de réponse, sûreté et protection des données.

Exemple:

Un extrait du cahier de charges d'une application de location des livres précise que:

- Le client peut :
 - ⇒ Louer et rapporter des livres.
 - ⇒ Les locations sont enregistrées ainsi que les clients.
- Le gérant peut:
 - ⇒ Ajouter / Supprimer des livres au catalogue.
 - ⇒ Changer les prix des livres au tarif.
- L'application doit:
 - ⇒ Calculer le prix d'une location selon le tarif des livres empruntés et la durée de l'emprunte.

2. Validation des besoins:

Une fois qu'un ensemble de besoins a été définies, ceci doivent être revues et validés afin de palier à la propagation des erreurs de cette phase au phase ultérieure du processus de développement.

3. Prototypage:

Il est peut être difficile de comprendre la nature exacte d'un problème, particulièrement, si le système est nouveau et aucun système non automatique n'existe pour servir de modèle au logiciel à développer.

Il est alors primordiale de développer une esquisse avant d'implémenter strictement les fonctionnalités du produit et d'ignorer les spécifications non fonctionnelles de ce que sera le système final.

L'esquisse fait l'objet de discussion entre les développeurs et l'utilisateur afin d'approuver, compléter et reformuler les besoins et d'éliminer les incohérences (les contradictoires).

- Un prototype doit:
 - ⇒ Mettre en évidence les incompréhensions (développeur / utilisateur)
 - ⇒ Détecter les oublies.
 - ⇒ Découvrir les contradictions.
 - ⇒ Evaluer rapidement la faisabilité.
 - ⇒ Servir de base à l'écriture de spécification complète et cohérente.
- Types de prototype:
 - ⇒ La maquette: est un système incomplète dont l'aspect extérieur est le même que celui du système final à réaliser. Elle est destinée à tester l'efficacité du produit final d'approuver les dialogues Homme/Machine.
 - ⇒ Le prototype jetable: après avoir étudié la faisabilité du produit grâce à un prototype, on jette celui-ci et on recommence le développement.
 - ⇒ Le prototype évolutif: La première version du prototype devient l'embryon (foetus) du produit final. On itère jusqu'au produit final.

II- La spécification:

A pour but de formaliser avec des outils, le cahier des charges et modéliser le système selon les différentes vues: données, traitement, métier (utilisateur),...

La spécification est une période du processus de développement pendant laquelle on s'exprime à un haut niveau d'abstraction et consiste en la définition des éléments logiciels qui réaliseront les fonctionnalités spécifiées pendant l'analyse des besoins. Les éléments ainsi définis doivent être conçus de manière à ce qu'il puisse être ultérieurement encodés dans un langage de programmation.

1. Techniques et outils de spécification:

- ⇒ Les spécifications informelles: utilise un énoncé structuré (chapitre, section, paragraphe,...) et exprimé en langage naturel (règles de gestion par exemple).

Elles sont très facilement communicables à des spécialistes, mais elles augmentent le risque de l'incohérence des spécifications.

- ⇒ Les spécifications formelles: la syntaxe et sémantique (structure et lexique) sont définies formellement par les outils mathématiques: Langage VDM, Z,...)

Elles sont difficiles à apprendre mais sont facile à encoder (possibilité d'avoir des compilateurs).

- ⇒ Les spécifications semi formelles: elles favorisent la communication entre développeur et utilisateurs en utilisant une représentation graphique des données et des traitements.

Le langage graphique doit être normalisé, et non sémantique non ambiguë (double, copie). Toutefois, les diagrammes sont en général accompagnés des textes écrits en langage naturel.

II-1 Des techniques de spécification pour les phases d'analyse

- ⇒ **Modèle Entité/ Association (CHEN):**

Il est bien adapté à la conception et modélisation les bases des données relationnelles.

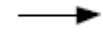
- ⇒ **Les diagrammes de flots de données (DFD):**

Ils conviennent pour la spécification des traitements. Les DFD montrent comment chaque processus (fonction) transforme ses entrées en sorties (flot entrant, flot sortant). Les DFD sont généralement complétés par un diagramme de contexte qui permet de faire figurer les acteurs extérieurs intervenants sur le système.

- Eléments syntaxiques:



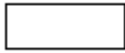
: Les fonctions.



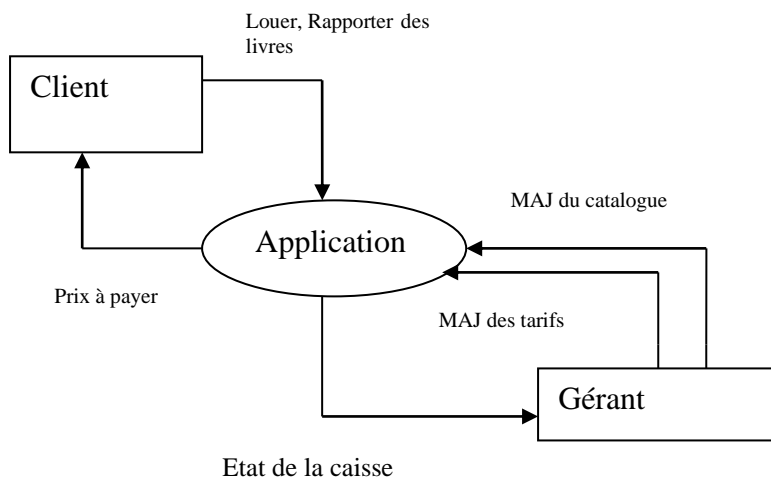
: Les flots.



: Les stockages.



: Les entités externes.

**Diagramme de contexte.**