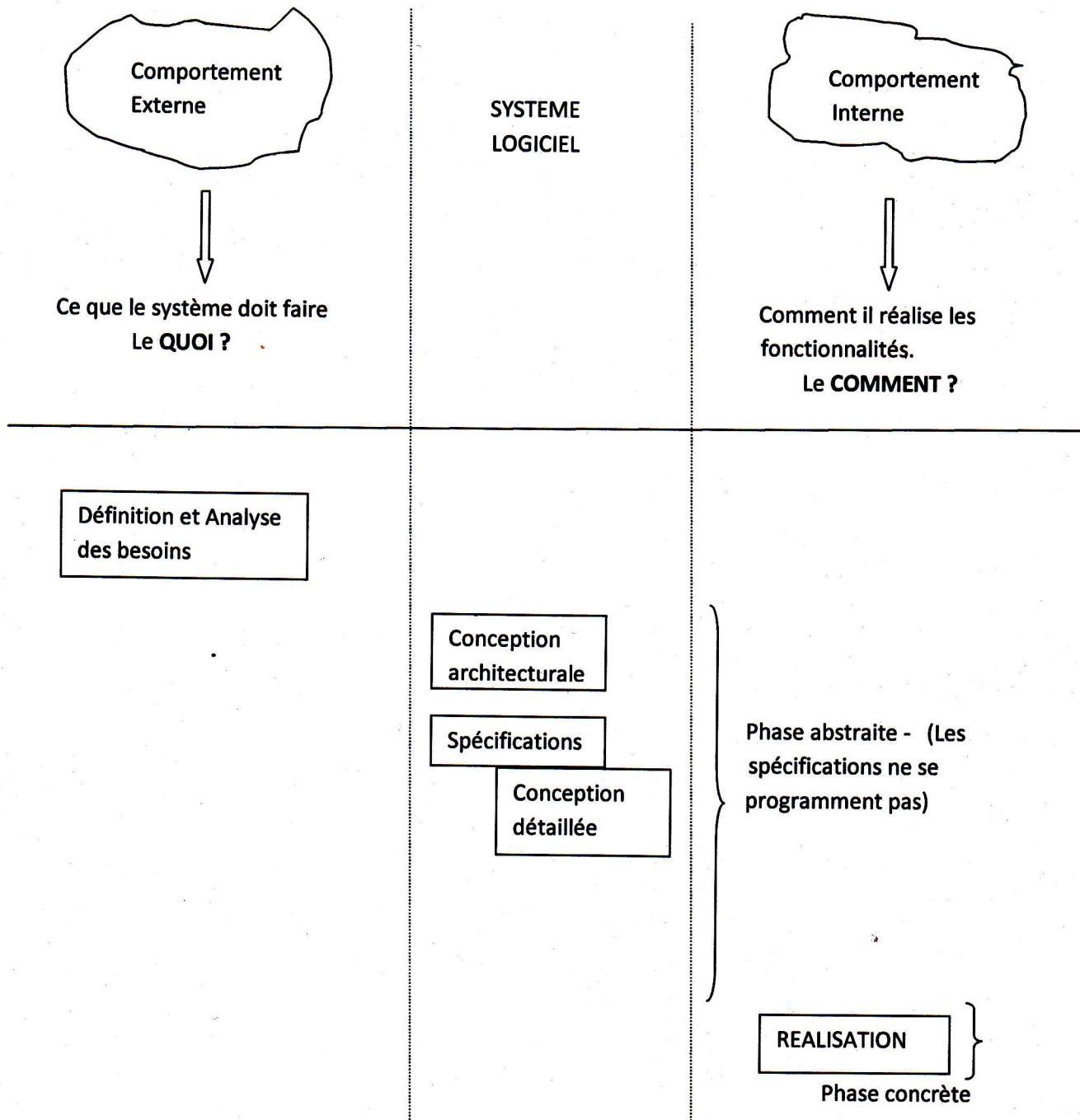


Chapitre II Spécifications des logiciels

1- Introduction

Tout système logiciel peut être décrit suivant un *comportement interne* et un *comportement externe* (Figure ci-dessous).



- a- Le comportement externe d'un système logiciel indique ce que ce système doit faire (pour répondre aux besoins) - **le Quoi ?**

Le Quoi ? d'un système logiciel correspond à la définition des besoins exprimés dans le cahier des charges (1^{ère} étape du cycle de vie) et qui se traduit par les fonctionnalités attendues du système.

- b- Le comportement interne d'un système logiciel indique, par contre, comment le système réalise ses fonctionnalités (**le Comment ?**).

Ce comment ? d'un système logiciel correspond à l'étape de « réalisation » (phase concrète) qui détaille « comment » le système fonctionne (code source, instructions, ...).

Le passage du quoi ? au comment ? d'un système logiciel nécessite une phase dite « étape de conception ». Cette étape de conception, considérée comme une phase abstraite, est nécessaire dans la mesure où c'est à ce niveau que l'on procède à la conception, d'une part, de l'architecture du système logiciel à développer, en termes de composants de cette architecture, et d'autre part au détail de chacun de ces composants (algorithmes, structures de données utilisées, ...).

2- Etape de conception des logiciels

La première étape du cycle de vie des logiciels consiste à faire l'analyse des besoins, c'est-à-dire les services que doit fournir un système logiciel à réaliser (besoins fonctionnels) ainsi que les contraintes sous lesquelles ce logiciel devra fonctionner (besoins non fonctionnels).

L'étape suivante consiste alors en la phase de conception du logiciel.

Pour de petits systèmes (logiciels de petite taille –Milliers de lignes de codes), il est facile d'aller directement de l'analyse des besoins à la conception détaillée des composants du logiciel, en termes d'algorithmes et de structures de données associées. **Ce qui n'est pas le cas** de gros logiciels (plusieurs dizaines de Milliers ou Millions de lignes de code, par exemple) où la phase de conception devra se faire en plusieurs étapes, principalement l'étape de conception générale (architecturale) et l'étape de conception détaillée.

- l'étape de **conception globale ou architecturale** consiste alors à déterminer l'architecture globale d'un système logiciel en termes de composants (modules, classes, ...), ainsi qu'en termes de relations entre ces composants.
- L'étape de **conception détaillée** des composants se traduit par l'expression des algorithmes relativement aux différents composants (algorithmes des méthodes définis dans un module ou une classe, par exemple) et des structures de données utilisées.

Pour des systèmes logiciels complexes, Il est très souvent UTILE (voire nécessaire) de compléter ces deux étapes, par une étape dite de « spécification des logiciels ». Cette étape permet de mieux préciser le rôle de certains composants complexes ; ce qui aura pour conséquence d'aider à une meilleure conception de ces composants, d'une part, et de faciliter les tests ultérieurs, d'autre part, assurant ainsi une meilleure maintenabilité.

3- Définition générale d'une spécification

Le terme de « spécification », vu comme une partie de la phase de conception d'un logiciel, correspond à une définition abstraite des composants logiciels, en termes de domaines d'entrées et de domaines de sorties des différents composants, avec précision des contraintes à satisfaire aussi bien en entrée qu'en sortie.

Exemple : Soit à définir un composant, dans ce cas une simple fonction, correspondant à la définition du calcul factoriel F d'un nombre entier N.

Une première spécification de cette fonction pourrait être :

Domaine d'entrée :

Entrée : Nombre entier

Condition (sur l'entrée) : Entier positif ou nul, $N \geq 0$

Domaine de sortie :

Sortie : F entier positif

Condition (sur la sortie) : F correspond au factoriel de N

Cette spécification indique que la fonction du calcul du factoriel F doit avoir comme entrée un nombre entier et que la condition à respecter sur cette entrée est que le nombre entier N doit être positif ou nul (domaine d'entrée)

Cette même spécification indique également, dans son domaine de sortie, que la sortie attendue F doit être aussi un nombre entier positif et que la condition sur cette sortie, pour que la fonction soit considérée comme correcte, est que F doit correspondre exactement au calcul du Factoriel du nombre entier N.

Remarques sur cette spécification :

- 1- La spécification des logiciels est « orientée concepteur » et non « orientée utilisateur ».

Cela veut dire que la spécification ne concerne pas l'utilisateur « final » du logiciel, mais concerne le concepteur dont le rôle est de mieux préciser les aspects conceptuels du logiciel à concevoir en établissant des spécifications.

- 2- La spécification de l'exemple ci-dessus est IMPRECISE car elle ne précise pas que N est le nombre entier en entrée (domaine d'entrée).

3- La correction de cette imprécision donne alors la spécification précise suivante :

Domaine d'entrée :

Entrée : Nombre entier N

Condition (sur l'entrée) : entier positif ou nul, $N \geq 0$

Domaine de sortie :

Sortie : F entier positif

Condition (sur la sortie) : F correspond au factoriel de N

NB : Comme une spécification a pour but de mieux préciser le rôle d'une fonction ou d'un composant, il est important que cette spécification soit PRECISE dans tous ses aspects liés à ses entrées et ses sorties.

4- La spécification ci-dessus peut également s'écrire plus simplement :

Entrée : Entier N

Condition : $N \geq 0$

Sortie : F

Condition : $F = N!$ (F doit correspondre au factoriel de N)

5- La spécification ci-dessus est précise mais **INCOMPLETE** car elle n'indique pas **CE QU'IL FAUT FAIRE** si l'entier N, en entrée, est négatif.

En effet, cette action n'est pas définie si $N < 0$ (spécification inconsistante). On peut alors compléter cette spécification comme suit :

Entrée : Entier N

Condition :

Sortie : F

Condition : $N \geq 0 \rightarrow F = N!$

$N < 0 \rightarrow$ écrire « calcul impossible »

- 6- Il est également possible d'écrire une spécification sans utiliser de variables ou objets informatique. Ainsi, la spécification ci-dessus peut s'écrire comme suit :

Entrée : Nombre entier

Condition :

Sortie : Nombre entier

Condition : Entrée ≥ 0 → Sortie = Entrée !

 Entrée < 0 → Sortie = « calcul impossible »

Exercices :

- 1- Etablir la spécification de la fonction « Factoriel » qui calcule le factoriel d'un nombre entier sachant que ce factoriel ne doit pas dépasser la valeur NMAX (capacité de l'ordinateur).
- 2- Etablir une spécification de la fonction de résolution d'une équation du second degré.