



# PLAN TYPE DE CAHIER DES CHARGES

Dernière révision : 26/02/07

UV1– INF301

2<sup>e</sup> semestre 2010

A. BEUGNARD, F. DAGNAT, J. MALLET ET S. ROUVRAIS

## Note relative à l'utilisation de ce document :

*Ce document apporte une aide à la caractérisation du besoin en fournissant la trame du contenu type d'un Cahier des Charges. Il aborde le besoin sous l'angle du service rendu au client.*

*Les textes en italique (comme celui-ci) sont des commentaires destinés à guider la rédaction du document final, ils seront de préférence soustraits de ce dernier pour plus de clarté. Ce document devra comporter toute information utile à son identification (nom du projet, date, auteur(s), num. de révision, etc.).*

*Le rédacteur a toute latitude pour adapter ce plan type au cas traité, par exemple en omettant certains paragraphes s'il estime qu'ils ne s'appliquent pas à son projet. Il doit cependant être en mesure de justifier ce choix. Enfin, l'attention du rédacteur est attirée sur le fait que ce document n'est pas un document de spécification, et qu'il ne doit pas s'y substituer.*

## Sommaire

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Objet du document . . . . .	3
1.2	Portée du document . . . . .	3
1.3	Terminologie . . . . .	3
1.4	Abréviations . . . . .	3
<b>2</b>	<b>Les objectifs du produit</b>	<b>3</b>
2.1	Définition du produit . . . . .	3

2.2	Contexte économique du produit . . . . .	4
2.3	Contexte d'exploitation du produit . . . . .	4
<b>3</b>	<b>Exigences sur le produit</b>	<b>4</b>
3.1	Capacités Fonctionnelles . . . . .	4
3.1.1	Description des fonctionnalités . . . . .	4
3.1.2	Interopérabilité . . . . .	5
3.1.3	Conformité réglementaire . . . . .	5
3.2	Exigences non fonctionnelles . . . . .	5
3.2.1	Fiabilité . . . . .	5
3.2.2	Sécurité . . . . .	5
3.2.3	Facilité d'utilisation . . . . .	5
3.2.4	Rendement . . . . .	6
3.2.5	Maintenabilité . . . . .	6
3.2.6	Portabilité . . . . .	6
3.3	Exigences concernant le développement du produit . . . . .	6
3.3.1	Objectifs de délais . . . . .	6
3.3.2	Objectifs de coûts . . . . .	6
3.3.3	Exigences de réalisation . . . . .	6
<b>4</b>	<b>Synthèse des exigences</b>	<b>7</b>
4.1	Hierarchisation des exigences fonctionnelles . . . . .	7
4.2	Hierarchisation des exigences non-fonctionnelles . . . . .	7
<b>A</b>	<b>Définition des Exigences qualité</b>	<b>7</b>

# 1 Introduction

## 1.1 Objet du document

Ce document décrit tous les services que doivent rendre le produit et ses livrables et **toutes** les exigences qu'ils doivent satisfaire.

## 1.2 Portée du document

Ce document est destiné à expliciter et formaliser le besoin du client YYY dans le cadre du projet XXX.

## 1.3 Terminologie

*Définir si nécessaire les termes spécifiques utilisés dans ce document.*

Terme	Description
Produit	Terme générique désignant l'objet de la demande du client. Il recouvre aussi bien un système qu'un service, sans préjuger de la part de logiciel, de documents et de matériel intervenant dans la réalisation.

## 1.4 Abréviations

*Définir si nécessaire les abréviations ou acronymes utilisés dans ce document.*

Abr.	Signification	Libellé
VIT	Vitale	Exigences fonctionnelles ou non fonctionnelles indispensables
IMP	Importante	Exigences souhaitées mais non exigées
MIN	Mineure	Exigences non exigées immédiatement, mais qui devront être prises en compte ultérieurement par le produit (impact sur l'évolutivité)

# 2 Les objectifs du produit

## 2.1 Définition du produit

*Présenter le produit, ses livrables et ses objectifs (mission, besoins commerciaux, économiques ...).*

## 2.2 Contexte économique du produit

*Présenter :*

- le marché sur lequel le produit se positionne ;
- la clientèle cible amenée à utiliser le(s) produit(s) ;
- éventuellement les zones géographiques cibles (France, Europe ...);

## 2.3 Contexte d'exploitation du produit

*Préciser :*

- les différents acteurs (nombre, rôle ...) amenés à utiliser ou à intervenir sur le produit ;
- la concurrence (p.ex. papyrus 2000 et Androsace 2005) ;
- les interfaces avec des produits ou services existants.

# 3 Exigences sur le produit

*Dans ce chapitre, les exigences seront décrites sans entrer dans le détail de la mise en œuvre (pas de sur-spécification).*

*Elles comprennent :*

- les capacités fonctionnelles (paragraphe 3.1), ce qui recouvrent l'ensemble des fonctionnalités (ou services) que doit offrir le système, ainsi que les aspects interopérabilité et conformité réglementaire ;
- les exigences non fonctionnelles (paragraphe 3.2). Il s'agit là de propriétés telles que la fiabilité, la sécurité, la facilité d'utilisation, la maintenabilité, etc. Ces exigences devront être décrites à la fois de façon qualitative (p. ex. le produit devra être très facile à utiliser par des utilisateurs novices) et de façon **quantitative** (p. ex. un utilisateur novice pourra maîtriser toutes les fonctions du produit en moins de 2 heures d'expérimentation). Il faut avoir le souci permanent d'être le plus précis possible. NB : En annexe, on trouvera la définition des exigences suivant la norme ISO/CEI 9126-1 : 2000, (Qualité des produits logiciels-Partie 1 : Modèle de qualité) ;
- les exigences liées au cycle de développement du produit (paragraphe 3.3), il s'agit notamment des objectifs de coûts et délais, du langage de programmation utilisé, de la documentation produite, etc.

*D'autres types d'exigences peuvent intervenir encore selon les spécificités du projet, comme par exemple : un protocole de certification du produit, des modalités d'exploitation, la fourniture de services associés, etc.*

## 3.1 Capacités Fonctionnelles

### 3.1.1 Description des fonctionnalités

*Décrire les grandes fonctions devant être satisfaites par le produit.*

*Il est souhaitable d'ordonner les fonctions par importance décroissante (en termes d'utilité) en établissant si possible des groupes fonctionnels.*

### 3.1.2 Interopérabilité

*Décrire la liste des interfaces du produit :*

- *environnement matériel et logiciel : outils et/ou fonctions logiciels ;*
- *citer les éventuelles contraintes de concurrence d'accès avec d'autres produits ainsi que les contraintes de réutilisation de produits existants.*

### 3.1.3 Conformité réglementaire

*Définir les normes, standards, lois et règlements auxquels doit se conformer le produit (loi informatique et liberté, etc.) et les limitations qui en découlent pour le concepteur dans le choix des solutions.*

## 3.2 Exigences non fonctionnelles

### 3.2.1 Fiabilité

*Décrire :*

- *le taux de défaillance maximum admis ;*
- *les contraintes de robustesse du produit. Pour chaque environnement prévu, décrire les agressions auxquelles il doit résister dans toutes les situations et ses réactions attendues ;*
- *décrire les exigences de récupération si le produit doit être capable de redémarrer seul après une défaillance de son environnement, etc.*

### 3.2.2 Sécurité

*Définir :*

- *le niveau de confidentialité des données manipulées et les différents niveaux d'accès aux fonctions et aux données ;*
- *les différents profils d'accès au produit et les droits correspondants ;*
- *les vulnérabilités du produit et les risques d'attaque, etc. ;*
- *les exigences de garantie de l'intégrité des données (procédures de sauvegardes automatiques, règles d'accès, etc.).*

### 3.2.3 Facilité d'utilisation

*Décrire les exigences concernant les interfaces homme machine, par exemple :*

- *facilité de compréhension du produit par l'utilisateur ;*
- *facilité d'apprentissage, durée d'apprentissage, de maîtrise du produit selon niveau initial, présence éventuelle d'un outil d'autoformation ;*
- *facilité d'exploitation (pour les administrateurs ou exploitants du produit) ;*

*NB : prendre en compte les différents types d'utilisateurs, qui peuvent être des administrateurs, des exploitants ou des utilisateurs finaux.*

### 3.2.4 Rendement

*Décrire les exigences :*

- de performances (temps maximum de réponse autorisé) ;
- de consommation de ressources (mémoire, volume disque requis) ;
- le nombre d'utilisateurs prévus ;
- le volume de données à manipuler et leur courbe de croissance prévisionnelle, etc.

### 3.2.5 Maintenabilité

*La maintenabilité peut être corrective (correction des erreurs) ou évolutive (amélioration du produit).*

*Les exigences concernent donc :*

- la facilité de correction ;
- la facilité à faire évoluer le logiciel ;
- les documents fournis pour accomplir ces tâches.

### 3.2.6 Portabilité

*Les exigences relatives à la portabilité du produit comprennent :*

- la facilité d'adaptation du produit à différents environnements (par exemple devra fonctionner sous GNU-Linux) ;
- la facilité d'installation (documents associés) ;
- La coexistence avec d'autres logiciels préexistants.

## 3.3 Exigences concernant le développement du produit

### 3.3.1 Objectifs de délais

*Ce sont les objectifs ou impératifs de délais qui impactent le développement du produit.*

### 3.3.2 Objectifs de coûts

*Ce sont les objectifs ou impératifs de coûts qui impactent le développement du produit.*

### 3.3.3 Exigences de réalisation

*Citer par exemple :*

- l'environnement cible ;
- les règles de programmation imposées ;
- le langage souhaité par le client ;
- une charte graphique imposée.

## 4 Synthèse des exigences

*Il s'agit d'établir une liste hiérarchique des fonctionnalités décrites au 3 par ordre d'importance décroissante. Ainsi on mettra en évidence l'importance relative de celles-ci, ce qui fournira un critère utile à la stratégie de réalisation : « réaliser en priorité des fonctions jugées les plus essentielles ».*

### 4.1 Hiérarchisation des exigences fonctionnelles

*La hiérarchisation consiste à classer l'exigence en lui attribuant un poids représentant son intérêt, utilité, degré d'importance.*

*L'objectif est de pouvoir ensuite fournir des éléments de compromis ou d'optimisation entre les différents paramètres de la solution technique. Ce critère peut aussi s'avérer très utile dans le cadre d'un développement incrémental.*

*Les exigences peuvent être évaluées, par exemple, comme vitales, importantes ou mineures.*

Nom fonction	Importance

### 4.2 Hiérarchisation des exigences non-fonctionnelles

Exigence	Importance

## Annexes

### A Définition des Exigences qualité

D'après ISO/CEI 9126-1 (2000) <sup>1</sup>.

Caractéristique	Sous caractéristiques	Définition : Attributs du logiciel portant sur :
-----------------	-----------------------	--

1. Notez que dans cette norme la propriété de sécurité est classée en exigence fonctionnelle.

<b>Capacité Fonctionnelle</b>	<i>Capacité du produit logiciel à fournir des fonctions qui répondent à des besoins exprimés et implicites lorsque le logiciel est utilisé dans des conditions spécifiées.</i>	
	Aptitude / Fonctionnalités	La capacité du produit logiciel à fournir un ensemble adéquat de fonctions pour les tâches et les objectifs spécifiés de l'utilisateur.
	Exactitude	La fourniture de résultats ou d'effets justes ou convenus.
	Interopérabilité	La capacité à interagir avec des systèmes donnés.
	Conformité réglementaire	La capacité du produit logiciel à respecter l'application de normes, de conventions, de réglementations de droit ou de prescriptions similaires.
	Sécurité	La capacité du produit logiciel à protéger les informations et les données de sorte que des personnes ou systèmes non autorisés ne puissent pas les lire ni les modifier et que les personnes ou systèmes autorisés y aient accès.
<b>Fiabilité</b>	<i>Capacité du produit logiciel à maintenir un niveau de service spécifié lorsqu'il est utilisé dans des conditions précises.</i>	
	Maturité	La capacité à éviter des défaillances dues à des défauts du logiciel.
	Tolérance aux fautes	La capacité à maintenir un niveau de service donné en cas de défaut du logiciel ou de violation de son interface.
	Possibilité de récupération	La capacité du produit logiciel à rétablir un niveau de service donné et de restaurer les données directement touchées en cas de défaillance.
<b>Facilité d'utilisation</b>	<i>Capacité du produit logiciel à être compris, connu, utilisé et à plaire à l'utilisateur, dans des conditions spécifiées d'utilisation.</i>	
	Facilité de compréhension	La capacité du produit logiciel à permettre à l'utilisateur de savoir si le logiciel est approprié et comment il peut être utilisé pour remplir des tâches particulières dans des conditions d'utilisation données.
	Facilité d'apprentissage	La capacité du produit logiciel à permettre à l'utilisateur d'apprendre son application.
	Facilité d'exploitation	La capacité du produit logiciel à permettre à l'utilisateur de l'exploiter et de contrôler son exploitation.
	Pouvoir d'attraction	La capacité du produit logiciel à attirer l'utilisateur.
<b>Rendement</b>	<i>Capacité du produit logiciel à fournir des performances appropriées en fonction de la quantité de ressources utilisée, dans des conditions données.</i>	
	Comportement vis-à-vis du temps	La capacité du produit logiciel à fournir des temps de réponse, de traitement et des débits appropriés lors de l'exécution de sa fonction, dans des conditions données.
	Utilisation des ressources	La capacité du produit logiciel à utiliser des quantités et des types de ressources appropriés lorsque le logiciel exécute sa fonction, dans des conditions données.



<b>Maintenabilité</b>	<i>Capacité du produit logiciel à être modifié. Les modifications peuvent inclure des corrections, des améliorations ou l'adaptation du logiciel aux changements d'environnement, d'exigences et de spécifications fonctionnelles.</i>	
	Facilité d'analyse	La capacité du produit logiciel à faire l'objet d'un diagnostic des déficiences ou des causes de pannes du logiciel, ou des pièces à modifier.
	Facilité de modification	La capacité du produit logiciel à permettre la mise en œuvre d'une modification spécifiée.
	Stabilité	La capacité du produit logiciel à éviter les effets inattendus des modifications du logiciel.
	Facilité de test	La capacité du produit logiciel à permettre la validation des modifications du logiciel.
<b>Portabilité</b>	<i>Capacité du produit logiciel à être transféré d'un environnement à un autre.</i>	
	Facilité d'adaptation	La capacité du produit logiciel à s'adapter à différents environnements spécifiés sans avoir recours à d'autres actions ou moyens que ceux prévus à cet effet pour le logiciel considéré.
	Facilité d'installation	La capacité du produit logiciel à être installé dans un environnement donné.
	Coexistence	La capacité du produit logiciel à coexister avec d'autres logiciels indépendants dans un environnement commun et en partageant des ressources.
	Interchangeabilité	La capacité du produit logiciel à être utilisé à la place d'un autre produit logiciel donné dans le même but et dans le même environnement.