

Objectifs d'apprentissage de la séance

Être capable d'identifier et de formaliser les besoins d'un client par rapport à un système logiciel selon les deux composantes suivantes :

- fonctionnalités du système telle qu'attendues par ses différents (futurs) utilisateurs ;
- aspects non fonctionnels à prendre en compte (fiabilité, rendement, facilité d'utilisation...).

Descriptif du problème et du travail à réaliser

La société BibTel souhaite se positionner sur le marché des services de réseaux sociaux. Elle lance pour cela un *appel à idées* en vue de faire réaliser un prototype de **mutualisation d'opinions** qu'elle entend baptiser *ToutAvis*, en référence à son implantation bretonne.

Ce système devra permettre aux usagers d'Internet de consulter et de donner leurs avis sur une gamme non limitée de produits et services (culture, gastronomie, etc.). BibTel souhaite proposer aux internautes un produit à caractère particulièrement participatif et collaboratif.

La réponse des soumissionnaires comprendra les éléments suivants :

1. un *bref descriptif* (quelques lignes) présentant les points forts de la proposition ;
2. un *diagramme des cas d'utilisation* montrant les fonctionnalités du système proposé selon le formalisme UML (voir par exemple <http://uml.free.fr/>) ;
3. un cahier des charges respectant le plan type imposé par BibTel joint en annexe de ce BE. Chaque rubrique y sera renseignée selon la norme de qualité ISO/CEI 9126-1 (voir cette même annexe). Ce plan type considère 5 *propriétés non-fonctionnelles*, mais BibTel se satisfera de 3 parmi celles-ci laissées au choix des soumissionnaires. Ces propriétés devront être décrites à la fois en termes qualitatifs et quantitatifs.

À l'issue de la séance, vous déposerez votre soumission¹ sur Moodle au format PDF, archivé zip si elle comporte plusieurs documents.

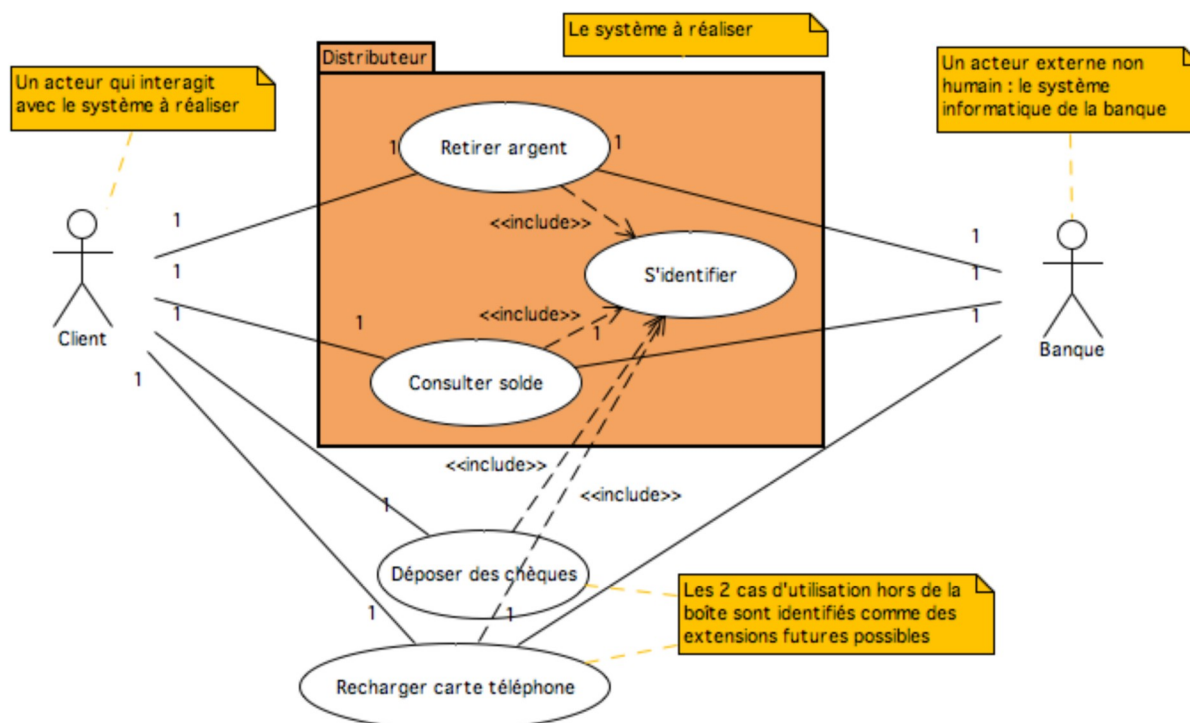
BibTel fera une synthèse de l'ensemble des propositions qui lui auront été faites et en tirera son propre cahier des charges. Elle proposera alors aux soumissionnaires de réaliser un démonstrateur conforme à celui-ci pour lui permettre de juger à la fois le concept et le savoir faire des soumissionnaires.

Démarche proposée pour produire votre livrable dans le temps imparti

1. Feuilletez le plan type de cahier des charges dans son intégralité pour identifier les rubriques à renseigner et estimez le volume de travail correspondant.
2. Examinez attentivement l'annexe de ce document intitulée « 5 DÉFINITION DES EXIGENCES QUALITÉS » pour vous permettre de bien comprendre le sens de chacune des rubriques à renseigner.
3. Établissez le point fort du produit que vous allez proposer et dressez la liste des services qu'il offrira (*capacités fonctionnelles* selon la terminologie du plan type). Attention : ayez une ambition à la mesure du temps dont vous disposez.

1 Vous trouverez sur [Moodle](#) le plan de cahier des charges au format ODT pour vous permettre de le renseigner. Le document que vous déposerez en retour sur Moodle devra impérativement être au format PDF.

4. Consignez ces services sous la forme d'un « diagramme de cas d'utilisation UML ». Pour sa forme², vous pourrez vous aider de l'illustration n°1. (Comptez une 20^{aine} de minutes)
5. Renseignez le formulaire de cahier des charges selon le plan type fourni. Pour ce faire,
 1. Renseignez d'abord la section 2 (Objectifs du produit) – durée 5mn.
 2. Allez directement ensuite à la section 3.2 (exigences non fonctionnelles). Choisissez parmi les cinq exigences proposées les trois qui vous paraissent les plus pertinentes et renseignez-les (ne pas oublier les aspects quantitatifs) – durée 30mn.
 3. Renseignez ensuite la section 3.3 – durée 10mn.
 4. Renseignez la section 3.1 (capacités fonctionnelles) *en conformité avec votre diagramme de cas d'utilisation*. Attention à bien interpréter les différents items (n'hésitez pas à soumettre un exemple à votre enseignant) – durée 30mn.
 5. Terminez par la section 1. Relisez intégralement votre document, corrigez-le le cas échéant (attention à la qualité de votre français).
6. Assemblez ou archivez les éléments de votre livrable et déposez-le sur Moodle au format demandé.



1. Illustration : Exemple de cas d'utilisation d'une application "terminal bancaire"

2 Note : Pour le saisir le diagramme des cas d'utilisation, vous pourrez utiliser le logiciel libre *Dia* et vous l'exporterez dans un format image pour l'intégrer à votre CDC

PLAN TYPE DE CAHIER DES CHARGES

Version 1.2

14/01/05

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

1 INTRODUCTION.....	3
1.1 Objet du document.....	3
1.2 Portée du document.....	3
1.3 Terminologie.....	3
1.4 Abréviations.....	3
2 LES OBJECTIFS DU PRODUIT.....	4
2.1 Définition du produit.....	4
2.2 Contexte économique du produit.....	4
2.3 Contexte d'exploitation du produit.....	4
3 EXIGENCES SUR LE PRODUIT.....	4
3.1 Capacités Fonctionnelles.....	4
3.1.1 Description des fonctionnalités.....	4
3.1.2 Interopérabilité.....	5
3.1.3 Conformité réglementaire.....	5
3.1.4 Sécurité.....	5
3.2 Exigences non fonctionnelles.....	5
3.2.1 Fiabilité.....	5
3.2.2 Facilité d'utilisation.....	5
3.2.3 Rendement.....	5
3.2.4 Maintenabilité.....	5
3.2.5 Portabilité.....	5
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
4 SYNTHÈSE DES EXIGENCES.....	6
5 DÉFINITION DES EXIGENCES QUALITÉ.....	7

1 INTRODUCTION

1.1 Objet du document

Ce document décrit **toutes** les exigences que doit satisfaire le produit.

1.2 Portée du document

Ce document est destiné à 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 et de matériel intervenant dans la réalisation.

1.4 Abréviations

Définir si nécessaire les abréviations utilisées dans ce document.

Abréviations	Libellé

2 LES OBJECTIFS DU PRODUIT

2.1 Définition du produit

Présenter les objectifs auxquels doit répondre le produit (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 ;
- les interfaces avec des produits ou services existants.

3 EXIGENCES SUR LE PRODUIT (voir annexe)

Dans ce chapitre, les exigences seront décrites sans entrer dans le détail (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 facilité d'utilisation, la maintenabilité, etc. Ces exigences devront être décrites à la fois de façon qualitative (ex : le produit devra être très facile à utiliser par des utilisateurs novices) et de façon **quantitative** (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 concernant le développement (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. Pour décrire chaque exigence fonctionnelle, on s'inspirera du tableau ci-dessous :

Nom	Libellé de la fonctionnalité, utiliser un verbe ; ex : « saisir un nouvel abonné ». Écarter impérativement les verbes vagues ou imprécis, ex : « gérer les utilisateurs »
Entrées	Les données ou informations en entrées nécessaires à l'exécution de la fonction ; ex : nom, prénom, date et lieu de naissance.
Événement déclencheur	Qui ou quel événement est à l'initiative du déclenchement de la fonction ; ex : l'opérateur, un échéancier implanté dans le système.
Sorties	Les données ou informations que la fonction doit produire en sortie ; ex : acquittement de bon déroulement.
Description	Le travail effectivement réalisé par la fonction ; ex : enregistrement des données associées au nouvel abonné. Préciser éventuellement les modalités utilisées (algorithme, etc.).
Contraintes	De façon générale, liste les conditions requises pour l'activation et la bonne exécution de la fonction ; ex : un abonné ne peut être enregistré qu'une seule fois .
Importance	Le niveau d'exigibilité de la fonction (indispensable, souhaitable, ... voir nomenclature §4.1)

3.1.2 Interopérabilité

Décrire la liste des interfaces du produit :

- environnement matériel et logiciel : outils et/ou fonctions logiciels ;
- interfaces avec les acteurs (définir les catégories d'utilisateurs), avec des fichiers ou des bases de données, avec d'autres produits ;
- citer les 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.

3.1.4 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 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 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.

3.2.3 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.

3.2.4 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.

3.2.5 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

Hierarchisation des exigences

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.

Abr.	Signification	Fonctions
VIT	Vitale	Indispensables
IMP	Importante	souhaitées mais non exigées
MIN	Mineure	Non exigées immédiatement, mais qui devront être prises en compte ultérieurement par le produit (<i>impact sur l'évolutivité</i>)

ANNEXE

5 DÉFINITION DES EXIGENCES QUALITÉ

D'après ISO/CEI 9126-1 (2000).

Caractéristique	Sous caractéristiques	Définition : Attributs du logiciel portant sur
Capacité Fonctionnelle	<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.
Fiabilité	<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.
Facilité d'utilisation	<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.
Rendement	<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.

Maintenabilité	<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.
Portabilité	<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.