

SUPPORTS RESSOURCES SAE 1.5 : Recueil des besoins**La boîte à outils !****Pourquoi recueillir les besoins de votre client ?**

Le recueil des besoins est une activité mettant en situation des acteurs qui ont des langages différents et qui n'ont pas les mêmes points de vue sur le produit à développer. Votre projet débute. Votre mission N° 1 consiste à bien cerner les besoins exprimés par le demandeur, appelé aussi client (ou utilisateur dans la plupart des cas).

C'est une phase primordiale qui permet de livrer une solution en rapport avec les attentes de ce dernier. Attention, **un demandeur a un besoin réel**, certes, mais il peut imaginer ce besoin et l'exprimer d'une manière complètement différente de ce dont vous avez l'habitude.

Pour garantir l'efficacité du recueil des besoins, vous devez traiter plusieurs sous-problèmes:

- Comment organiser une réunion de façon efficace avec le client, en faisant émerger ses besoins ?
- Comment traduire ces besoins en liste de fonctionnalités ?

Vous devez enfin produire un dossier d'expression des besoins couvrant plusieurs sujets :

- Objet et domaine d'application du projet
- Abréviations et terminologies utilisées dans le document
- Description des besoins par une liste de fonctionnalités, des maquettes, des scénarios...

Pour cela, vous allez avoir besoin d'étudier les méthodes et technologies de gestion de projet expliquées dans la partie « Ressources SAE ». Ce dossier d'expression de besoin sera plus tard transformé en **un cahier des charges** qui inclura plus de détails techniques sur les besoins ainsi que les contraintes du projet (cf. GPO2 en S2).

Remarques

*Chaque cas a ses particularités liées au **métier** du client. On doit toujours se poser une question : "Ai-je toutes les connaissances et les informations pour définir ce que doit faire le système à produire ? ". Dans un cycle de vie agile, le recueil des besoins n'est pas qu'une phase en amont du projet à l'issue de laquelle tous les besoins seraient recensés. C'est une activité itérative qui fait émerger les besoins de plus en plus précis au fur et à mesure que le client visualise les fonctionnalités implémentées.*

Ressources pour la SAE :**1. Comprendre la vision du projet**

La vision d'un produit ou d'un projet est l'orientation générale donnée à l'équipe, l'objectif global à atteindre, le « cap » à prendre. Sans objectif, sans vision partagée, l'équipe va errer sans savoir précisément où aller, dans quel but, ni comment y aller. Les retards sont pris en tout début de projet, parce que l'équipe n'a pas connaissance de cette vision ; elle est alors hésitante, refusant de prendre le risque de se fourvoyer dans une mauvaise direction... La production à l'issue de ce partage de vision permettra de répondre aux quatre questions suivantes :

1. **Quel contexte ?** Quel type d'organisation est votre client ? quel est son métier ? dans quel contexte apparaît la demande ?
2. **Pour quoi ?** Quel est l'objectif stratégique du client ?
3. **Quoi ?** Quels sont le périmètre et les contraintes, actuels et futurs, du projet ?
4. **Qui ou pour qui ?** Qui sont les parties prenantes du projet ? Qui va utiliser le produit ? A qui va bénéficier le produit ? A toute personne impactée ou qui peut impacter le déploiement d'un produit, y compris l'équipe de réalisation.
5. **Comment ?** Comment l'équipe va-t-elle réaliser le produit ? On pourrait proposer une brève ébauche de solution technique.

Un simple exercice vous facilitera la description de la vision. Vous rencontrez dans l'ascenseur quelqu'un que vous connaissez et qui travaille dans le domaine informatique. Il vous demande sur quel projet vous travaillez en ce moment et vous avez deux minutes pour lui donner une « vision » complète du projet. Vous pourrez employer un style d'expression comme suit :

- ☐ L'utilisateur ...
- ☐ ... a un besoin ...
- ☐ ... que le produit que nous développons, qui est un produit de type ...
- ☐ ... va satisfaire ainsi ...
- ☐ ... contrairement à d'autres, ou par rapport à la solution existante...
- ☐ ... donc se positionne de cette façon.

Un exemple concrète pour appliquer le style :

- ☐ Les jeunes diplômés niveau bac + 5...
- ☐ ... qui souhaitent postuler en ligne dans notre société, consulter des offres d'emploi et déposer des candidatures spontanées...
- ☐ ... disposent d'un site de recrutement, au travers d'un portail corporate...

❑ ... qui leur offre quotidiennement des offres mises à jour ainsi que des informations sur l'actualité de la société...

❑ ... contrairement à la presse quotidienne, moins rapide et pas interactive, notre site offre la possibilité de transmettre en temps réel leur lettre de motivation et leur CV ainsi que la possibilité de s'abonner aux offres en fonction d'un profil...

❑ ... c'est une application Web, accessible 24 h sur 24, actualisée quotidiennement, reflet du dynamisme de notre société et de sa proximité avec les jeunes diplômés.

2. Faire émerger des besoins

Un besoin de client est **une nécessité ou désir éprouvé** par un utilisateur, exprimé en termes de finalité, sans référence aux solutions techniques susceptibles d'y répondre. On doit distinguer le besoin initial du besoin à traiter. Il faut noter que ce qui doit être exprimé (Figure. 1) ne se limite pas à recenser ce qui est explicitement exprimé par le client, mais également à capter tout ce qui est implicite ou non dit: « une évidence » pour le client.

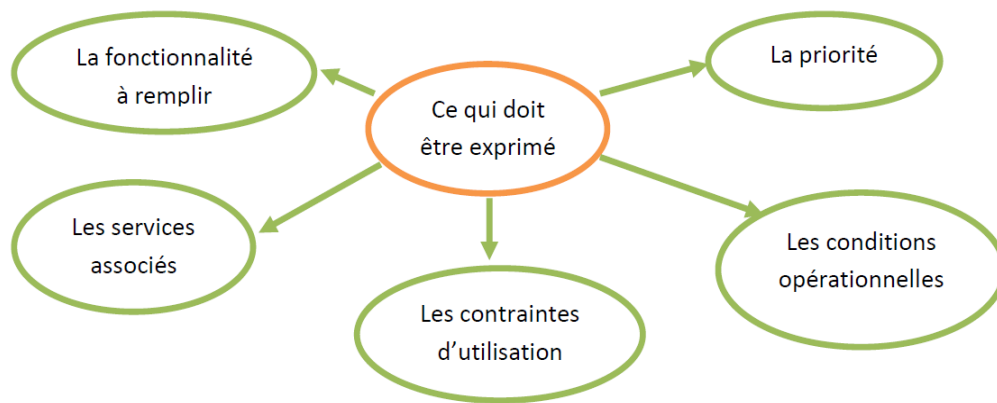


Figure. 1 Faire émerger les besoins

Remarques : Dans la gestion d'un projet agile, les besoins émergent toujours au fur et à mesure. Ainsi, le recueil et la hiérarchisation des besoins ne peuvent que s'inscrire dans une démarche itérative.

Outil : Technique des cases 9

	Quel est le problème ?	Qui est impacté ?	Visualiser la solution
Questions ouvertes : « Dites-moi... » « Racontez-moi... » « Et puis... »	1	4	7
Contrôle Combien ? Quand ? Où ?	2	5	8
Validation « Si je comprends bien... » - Si non, revenir aux questions ouvertes - Si oui, passer à la question suivante	3	6	9

Figure. (Tableau) 2 La technique des neuf cases

Le client commence par raconter des « histoires » sur son métier, ses difficultés. Il s'agit de questions ouvertes (1) pour explorer et comprendre. Des questions de contrôle (2) permettent de clarifier, d'apporter des précisions. L'interviewer doit ensuite valider sa bonne compréhension et synthétiser l'essentiel par la reformulation ; cette étape (3) permet de lever toute ambiguïté. Si la synchronisation s'opère (le client est compris par l'interviewer), ce dernier peut passer à l'étape suivante et tenter de comprendre qui est impacté par le problème et comment il est impacté (4). Le processus de contrôle et de validation est ensuite le même (5 et 6). Si la compréhension n'est pas totale, il doit revenir aux étapes 1 et 2. Lorsque le problème est bien identifié et les impacts mesurés, l'objectif suivant est d'amener le client à se représenter la solution idéale. Sans qu'il soit nécessaire, pour lui, d'élaborer la solution, il peut cependant visualiser un environnement où le problème n'existe plus.

3. Formaliser les besoins

Les besoins exprimés par les utilisateurs ou leur représentant sont formalisés dans un support documentaire linéaire. En tout état de cause, ils sont exprimés dans un langage métier, celui des utilisateurs, en termes d'usage ou de services attendus. Ces besoins sont ensuite pris en charge par l'équipe de réalisation puis introduits dans un cycle de fabrication (cycle de vie de réalisation du projet cf. GPO2) ; ce cycle de fabrication comporte une succession d'étapes au cours desquelles le besoin subit plusieurs transformations et revêt la forme d'un modèle, d'un diagramme, de lignes de code, d'un composant technique, d'un document, d'un scénario de test...pour devenir une fonctionnalité opérationnelle.

Les User story

En méthode Agile : Une étape peut être d'utiliser les **User Story** pour décrire les fonctionnalités attendues du système.

Cf. <https://blog.myagilepartner.fr/index.php/2017/08/02/user-stories-bien-les-rediger/>

La base de l'user story, c'est de commencer par décrire une demande fonctionnelle à partir d'un utilisateur clé (un persona cf.5) tout en écrivant avec un langage naturel qui permet à toute personne de comprendre la demande :

demande fonctionnelle classique : « le client doit pouvoir ajouter un produit dans son panier pour l'acheter »

le titre de la user story : « En tant que client, je souhaite pouvoir ajouter un produit dans mon panier afin de pouvoir l'acheter ».

On suit en fait un format strict pour que toutes les user stories suivent les mêmes règles ; voici le format à respecter :

En tant que [persona], je souhaite [souhait] afin de [but]

Bref une UserStory tient sur un Post-It !

Le diagramme de cas d'utilisation

Vous pouvez utiliser le diagramme de cas d'utilisation UML pour décrire le système à développer, d'un point de vue strictement fonctionnel : les échanges entre le système à développer et les acteurs externes (utilisateurs ou d'autres systèmes). Il faut savoir qu'un cas d'utilisation ne décrit que les exigences fonctionnelles, qu'il faudra compléter avec d'autres exigences : interfaces externes, règles métier, formats de données, performance... De plus, un diagramme de cas d'utilisation UML peut être complété par une description textuelle dont le modèle est illustré comme ci-dessus. Un exemple concret pour décrire un cas d'utilisation de retrait d'agent par un guichetier à la banque, correspondant au diagramme illustré dans la Figure. 3. Dans l'exemple, la partie « Scénario nominal » peut être également décrit par un diagramme d'activité UML.

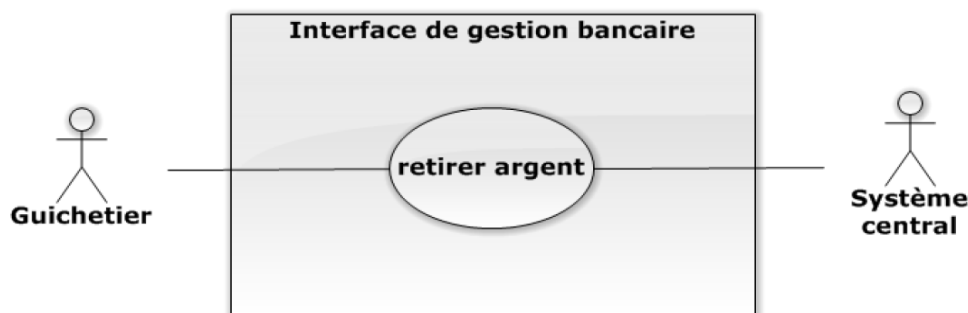


Figure. 3 Diagramme de cas d'utilisation

DESCRIPTION TEXTUELLE DU CAS D'UTILISATION

Cas d'utilisation : *Retrait d'argent*

Acteur : Guichetier

Événement déclencheur : Lors qu'un client demande le retrait volumineux d'espèces

Parties prenantes et intérêts : Client, guichetier, système central de banque

Niveau : Objectif utilisateur

Portée: Au guichet de la banque

[Pré-condition :] Le client possède un compte (donne son numéro de compte et montre une pièce d'identité valide)

Scénario nominal :

1. Le guichetier saisit le numéro de compte client.
2. L'application valide le compte auprès du système central.
3. L'application demande le type d'opération au guichetier.
4. Le guichetier sélectionne un retrait d'espèce du montant demandé.
5. L'application demande au système central de débiter le compte.
6. Le système notifie au guichetier qu'il peut délivrer le montant demandé.

[Post-condition :] Le guichetier ferme le compte. Le client récupère l'argent.

[Extensions :] Aucune

[Contraintes non fonctionnelles :] Toute l'opération doit se dérouler avec confidentialité.

[Questions-ouvertes et annexes :] Aucune

4. Hiérarchiser les besoins

L'un des motifs d'échec de nombreux projets réside dans le fait que les fonctionnalités ne sont pas développées par ordre de priorité. Si l'on admet que l'on a rarement assez de temps pour tout faire, que les changements surviennent inévitablement au cours du projet, il faut, dès lors, qualifier et valoriser les besoins pour les prioriser afin de faciliter les arbitrages sur les modifications de périmètre.

Modèle de Kano On ne doit pas présumer de l'importance de telle ou telle fonctionnalité pour l'utilisateur ; son appréciation est essentielle. Par conséquent, il est fortement conseillé de mener une enquête auprès d'un échantillon représentatif d'utilisateurs pour obtenir cette échelle de valeur, en leur posant deux questions sur chacune des fonctionnalités :

1. « Que pensez-vous du produit s'il contient cette fonctionnalité ? » --- Question fonctionnelle
2. « Que pensez-vous du produit s'il ne contient pas cette fonctionnalité ? » --- Question dysfonctionnelle

Alors, nous disposons de cinq réponses possibles :

1. « Cela me ferait plaisir. » --- Plaisir
2. « Ce serait un minimum. » --- Minimum
3. « Je n'ai pas d'avis. » --- Neutre
4. « Je l'accepterais. » --- Acceptation
5. « Cela me dérangerait. » --- Dérangement

En fonction des réponses obtenues, vous pouvez établir un tableau (Figure. 4) indiquant le statut des fonctionnalités en fonction de la combinaison des réponses aux deux questions.

Question fonctionnelle		Question dysfonctionnelle				
		Plaisir	Minimum	Neutre	Acceptation	dérangement
	Plaisir	?	L	L	L	E
	Minimum	R	I	I	I	O
	Neutre	R	I	I	I	O
	Acceptation	R	I	I	I	O
	dérangement	R	R	R	R	?

O Obligatoire R « Reverse » (Inversée)

E Exprimée ? Incertaine

L Latente I Indifférente

Figure 4. Statut des fonctionnalités

On distingue alors six catégories :

1. **Exigence obligatoire** : qui doivent impérativement être satisfaites. Une impasse sur ces exigences créera une frustration et un mécontentement chez le client.
2. **Exigence exprimée** : sa satisfaction est proportionnelle au niveau de performance et de coût de la fonctionnalité : si la fonctionnalité ne répond pas précisément à son besoin, le client sera insatisfait.
3. **Exigence latente** : qui correspond à des besoins émergents, pas vraiment exprimés, qui crée « l'heureuse surprise » et ont un effet très positif sur la satisfaction du client.
4. **Exigence indifférente** : qui ne préoccupe pas l'utilisateur.
5. **Exigence incertaine** : dont l'utilité ne peut être démontrée aisément à l'utilisateur.
6. **Exigence inversée** : il s'agit d'une fonctionnalité dont le client attend l'inverse de celle proposée.

Dès que vous avez les réponses (le tableau dans la Figure. 4), vous pouvez l'utiliser pour déterminer l'ordre d'implémentation des fonctionnalités : toutes les exigences obligatoires doivent être implémentées, le maximum d'exigences exprimés ainsi que quelques exigences latentes, pour séduire le client (*dès les premières livraisons dans un cycle de vie de développement agile !*).

5. Maquettage à l'aide de PERSONAE

Source : <https://www.weloveusers.com/partage/cours/apprendre-personas.html>

Les personas sont des **outils d'aide à la conception** et de communication. Ils permettent à l'équipe de conception de rester centrée sur les utilisateurs finaux (*end-users*), leurs aptitudes et leurs buts. Ils permettent une conception réaliste basée sur ce que l'utilisateur final peut et veut faire, ils sont donc une aide pour les User Story.

Une autre pratique consiste à combiner les personas à des parcours utilisateurs (user flow). Il s'agit de représenter schématiquement (diagramme, storyboard, etc.) le parcours emprunté par l'utilisateur pour atteindre son objectif.

Exemple :

- Pour l'interface d'un logiciel, le parcours pourrait être une description de la séquence d'écrans que l'utilisateur consulte et les choix qu'il effectue dans des menus de navigation.

