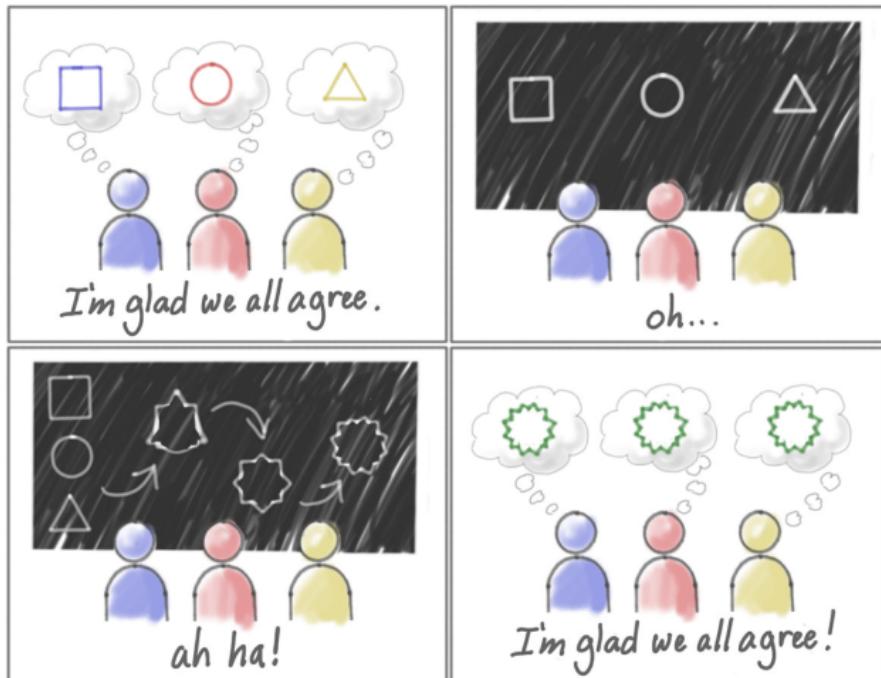


Résumé et inspiré du livre User Story Mapping par Jeff Patton

**L'idée :** Une story c'est *write something on a card, talk about it, and agree on what to build*

## 1 Bien se comprendre ?



Si j'écris une idée que j'ai derrière la tête et que je vous demande si vous êtes d'accord alors vous allez tous me répondre "oui". Cependant, si nous nous rencontrons et discutons, vous pourrez me dire ce que vous pensez et je pourrai vous poser des questions. Chacun d'entre nous va comprendre des choses différentes, voir des aspects importants différents. Nous exposons aux autres nos pensées pour parvenir à une compréhension commune. Nous nous sentons maintenant alignés et confiants dans le fait que nous avançons ensemble.

### 1.1 Discuter au lieu de rédiger une documentation

**Affirmation 1.** *Le but réelle d'une user-story est la compréhension.*

Lorsque des personnes lisent des documents et en retirent des compréhensions différentes, c'est soit la faute du lecteur, soit celle du document. A la place, privilégier des conversations avec des mots, des images pour construire une compréhension commune.

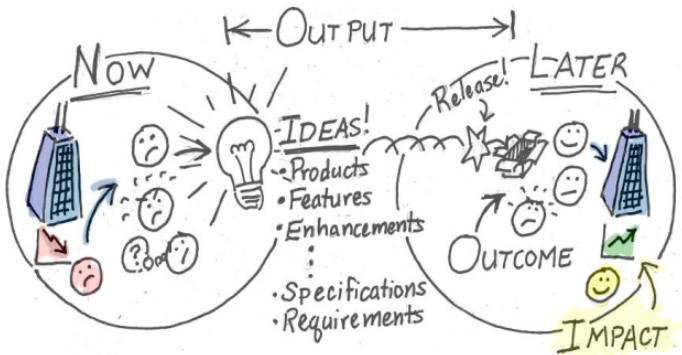
### 1.2 Des documents pour se rappeler

Lorsqu'on raconte une histoire pas besoin de rédiger un roman, quelques notes, dessins, stickers pris en photo suffisent pour que l'ensemble de l'équipe se rappelle de l'échange.

**Affirmation 2.** *Les choses les plus importantes ne sont pas celles écrites mais celles que nous nous rappelons lorsqu'on les voient.*

### 1.3 Le logiciel est une des finalités

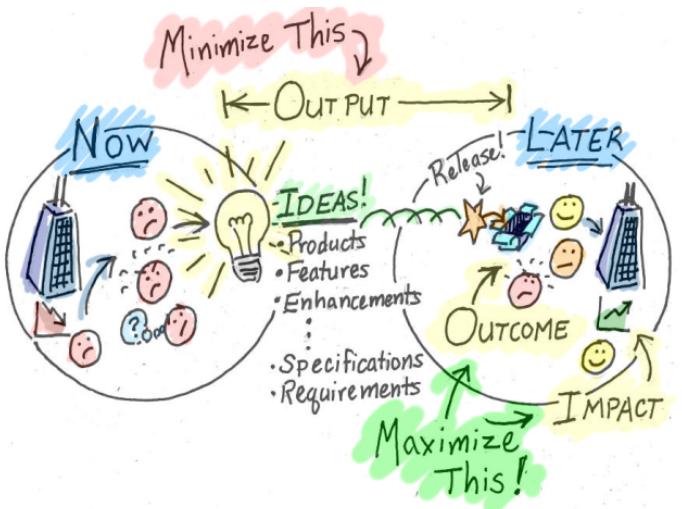
**Affirmation 3.** Une bonne histoire concerne les "pourquoi" et les "pour qui", et non juste les "quoi".



Le logiciel est le résultat final. Lorsqu'on crée un projet, on souhaite créer quelque chose de novateur, qui doit "changer le monde". Quand on parle les fonctionnalités de notre produit on va parler du *pour qui* et *pourquoi* on le fait.

### 1.4 Construire moins mais mieux

**Affirmation 4.** Dans le développement logiciel on a toujours plus à construire que nous n'avons le temps et les ressources alloués.



A la fin d'une journée, nous devrions minimiser le output (tout ce qu'il y a entre maintenant et la livraison) et maximiser le gain. Il faut être très attentif aux personnes à qui on essaie de résoudre les problèmes. Que ce soit ceux qui achètent votre logiciel ou ceux qui vont utiliser votre logiciel.

### 1.5 Que retenir ?

- Les stories ne sont pas une forme écrite des exigences. Raconter des stories avec les mots, des schémas permet une meilleure compréhension.
- Les stories ne sont pas des exigences, ce sont des discussions sur la résolution de problèmes de notre organisation, de nos clients et de nos utilisateurs qui conduisent à des accords sur ce qu'il faut construire.
- Votre travail ne consiste pas à créer plus de logiciels plus rapidement : il consiste à maximiser le résultat et l'impact que vous obtenez de ce que vous faites.

*The best solutions come from collaboration between the people with the problems to solve and the people who can solve them.*

## 2 User story map

**Definition.** L'user map permettent de décomposer les grosses user-stories au fur et à mesure qu'on l'a raconte

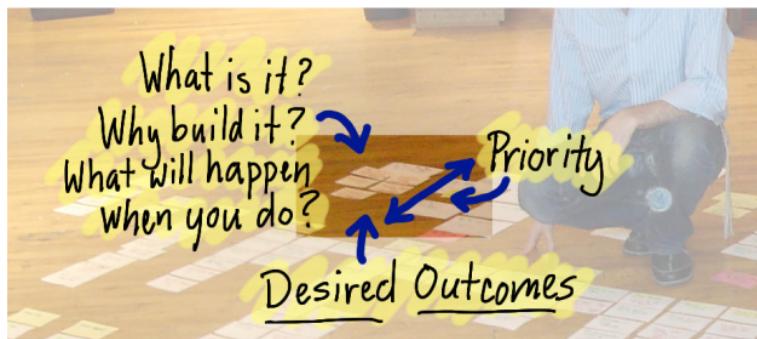
L'équipe entière va se rendre compte de problématique sous-acescente et cela va entraîner la création de nouvelles users-stories.

### 2.1 Écrire des cartes

**Definition.** Parlez et documentez : écrivez des cartes ou des post-it pour extérioriser votre pensée lorsque vous racontez des histoires.

Afin de se rappeler des idées évoquées, nous allons les retranscrire sur un tableau en quelque mot en suivant le patron *Think — Write — Explain — Place*

1. Rédiger quelques mots sur un post-it après avoir pensée à l'idée
2. Expliquer son idée : mots, gestes, dessins. On raconte une histoire
3. Placer la carte



### 2.2 Réorganiser les cartes

**Definition.** Réorganiser les cartes ensemble vous permet de communiquer sans dire un mot.

- Naturellement si on place une carte au dessus d'une autre alors cela signifie qu'elle est plus importante.
- Naturellement si on positionne une carte à gauche puis une à droite cela signifie une séquence.

**Definition.** Cartographier nos stories nous aide à voir les trous

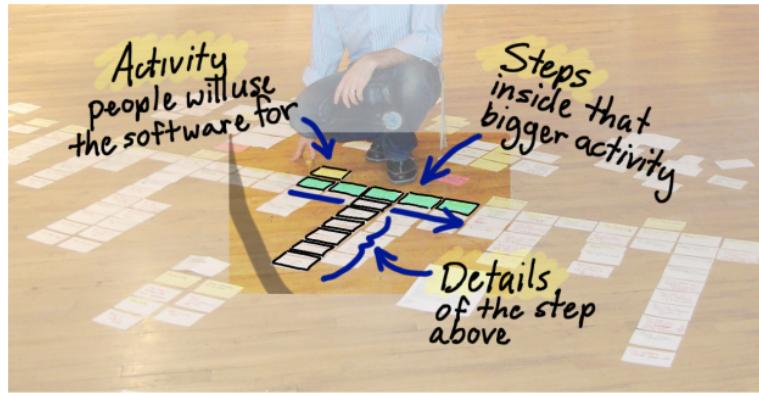
On partage sa compréhension, et les autres vont comprendre si nous avons le même point du vue du problème.



## 2.3 Explorer les détails et options

Les cartes que nous plaçons en haut de chacune des colonnes deviennent de *big things*. Puis les détails se répartissent en dessous. Nous nous arrêtons à chaque étape de la story et posons les questions suivantes :

- Quelles sont les choses spécifiques qu'on fera ici ?
- Quelles sont les autres choses qu'on pourrait faire ?
- Qu'est-ce qui rendrait la chose vraiment cool ?
- Qu'en est-il lorsque les choses tournent mal ?

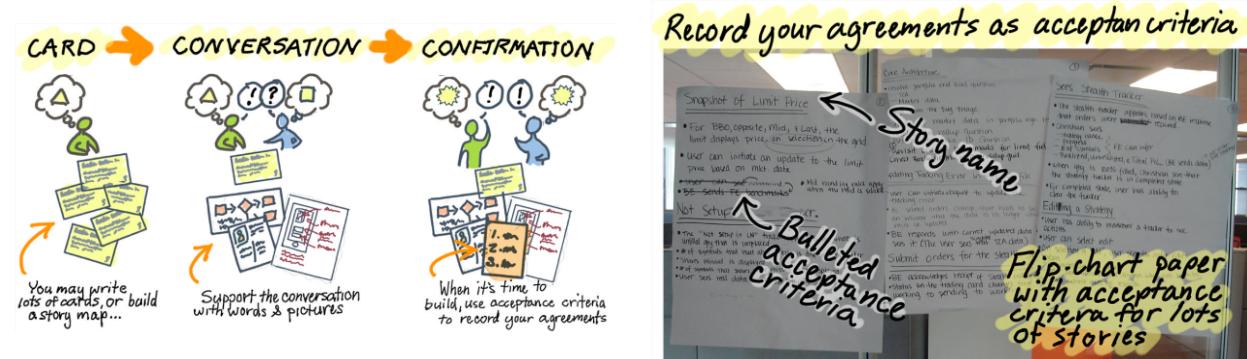


## 3 Mise en oeuvre

Une *User story* comportent trois aspects essentiels. Nous pouvons les appeler Carte, Conversation et Confirmation.

**Affirmation 5.** *La carte est un mémo sur les discussions menées et les accords passés.*

### 3.1 Card, Conversation, Confirmation

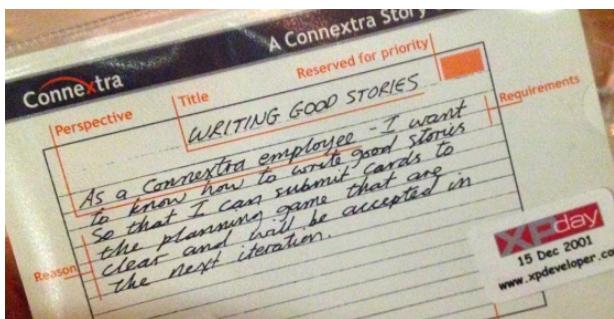


**Affirmation 6.** *Les bonnes conversations de story ont beaucoup de mots et d'images*

**Affirmation 7.** *Conserver nos accord (confirmation) en écrivant des critères d'acceptance*

La discussion va nous pousser à répondre aux questions et cette liste de "choses" va devenir nos critères d'acceptance.

## 3.2 Comment écrire une carte



En tant que [utilisateur]  
Je veux [faire]  
Afin que [obtenir du bénéfice]

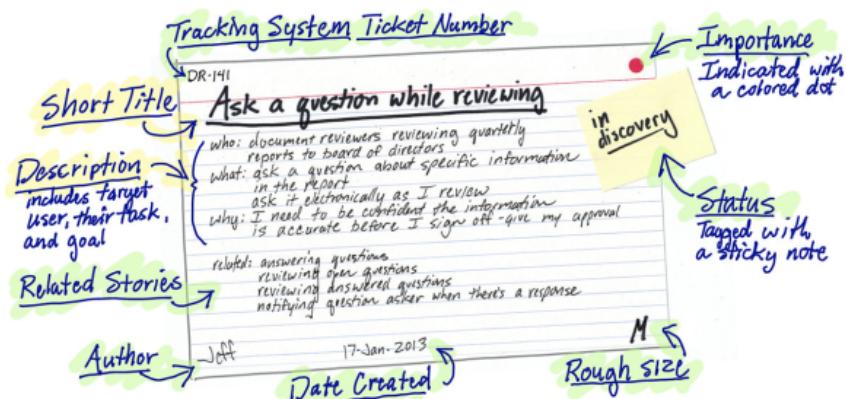
La story doit nous forcer à penser aux *qui*, *quoi* et *pourquoi*. Elle permet ainsi de commencer la conversation

En tant que manager d'un groupe de musique  
Je souhaite télécharger une image  
Afin de personnaliser mon flyer promotionnel.

**Objectif :** La carte permet de commencer la discussion, *pourquoi veut-il personnaliser son flyer ?*, *Qui va héberger les photos*, etc ...

Avec une carte on doit répondre à minima aux questions.

- Pour qui : à qui s'adresse la fonctionnalité, quel utilisateur précisément.
- Quoi : qu'est-ce qui est attendu
- Pourquoi : fait-on cette fonctionnalité. Il faut donner du sens au travail à réaliser.



Les post-it permet de discuter rapidement d'un sujet. Les cartes permettent de résumer notre pensée afin de pouvoir commencer à travailler dessus.

- Titre
- Description : répond aux questions *qui*, *quoi* et *pourquoi*
- Estimation/Budget
- Valeur ajoutée
- Critères d'acceptance : on peut les écrire au dos de la carte

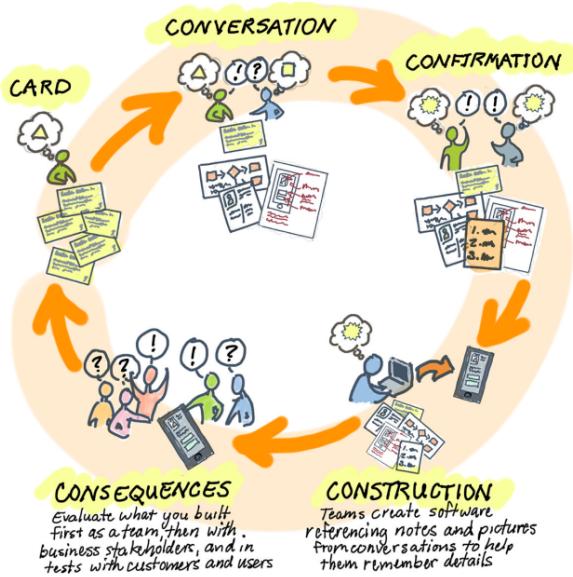
## 3.3 Raconter de meilleures histoires

La conversation va nous permettre d'écrire notre carte. Nous devons y aborder les différents sujets qui suivent :

- Qui : évoquer précisément de qui il s'agit. Il peut exister différents types d'utilisateur avec différents points de vue.

- Quoi : C'est ce que veulent les utilisateurs. Parler de l'interface utilisateur (comment doit-elle être), parler des différents services (e.g. moyen de paiement).
- Pourquoi : pourquoi l'utilisateur veut ce produit/fonctionnalité. Trouver les raisons sous-jacentes. Pourquoi les utilisateurs seraient impliqués, pourquoi les parties prenantes seraient impliquées.
- En dehors du logiciel :
  - L'endroit où se trouvent les utilisateurs de votre produit lorsqu'ils l'utilisent.
  - Du moment où ils utilisent le produit et de la fréquence à laquelle ils l'utilisent.
  - Qui d'autre est présent lorsqu'ils l'utilisent.
  - ⇒ Tous ces éléments donnent des indices sur ce que pourrait être une bonne solution.
- Ce qui ne va pas
  - Que se passe-t-il lorsque les choses tournent mal ?
  - Que se passe-t-il lorsque le système est en panne ?
  - Comment les utilisateurs pourraient-ils faire autrement ?
  - Comment répondent-ils à leurs besoins aujourd'hui ?
- Questions et hypothèse : Identifiez vos questions et discutez de l'importance d'obtenir une réponse avant de créer un logiciel. Prenez le temps de remettre en question vos hypothèses.
  - Comprenez-vous vraiment vos utilisateurs ?
  - Est-ce vraiment ce qu'ils veulent ?
  - Ont-ils vraiment ces problèmes ?
  - Vont-ils vraiment utiliser cette solution ?
  - Existe-t-il des risques techniques à prendre en compte ?
- Imaginer la solution : La grande victoire se produit lorsque les participants à une conversation sur une histoire posent quelques hypothèses originales sur ce que la solution devrait être.
- Le Comment : même si nous sommes dans un processus de réflexion (*quoi et pourquoi*) si une personne qualifiée intervient sur le *comment* il faut la laisser parler.
- De la durée : nous devons prendre des décisions pour aller de l'avant avec la construction de quelque chose ou non. Dans les premières conversations, cela peut se traduire par "beaucoup de temps" ou "quelques jours". Il est encore mieux de le comparer à quelque chose de déjà construit : "à peu près la même chose que la fonction de commentaire que nous avons créée le mois dernier". Il s'agit d'une estimation et non d'un engagement, au fur et à mesure des discussions nous allons accroître la précision.

### 3.4 Utiliser les cartes



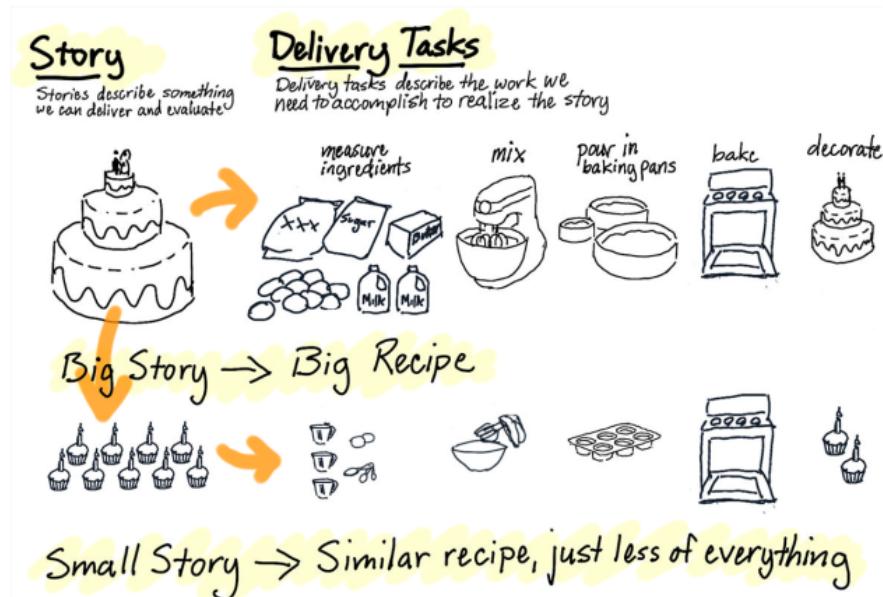
Les cartes ont permis de **construire une vision claire et commune**. Après avoir rédiger les détails, les critères d’acceptance nous sommes prêt pour développer le produit.

Une fois le travail fini nous allons pouvoir faire une rétrospective du travail. Expérience utilisateur, qualité fonctionnelle, qualité du code, etc ... Nous tirons les conséquence de notre travail.

## 4 La taille des stories

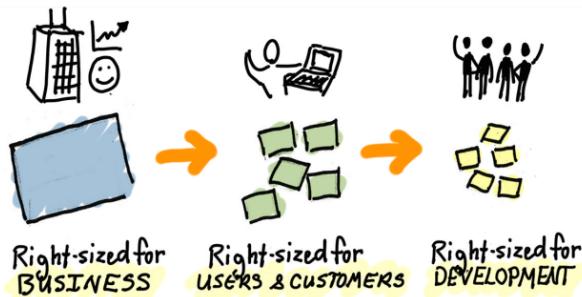
**Affirmation 8.** Si la story décrit une solution trop coûteuse, envisagez une solution différente qui vous aide à atteindre l’objectif.

**Affirmation 9.** Si la story décrit une solution abordable mais trop grosse, décomposez-la en parties plus petites qui vous permettront d’évaluer et de voir les progrès plus rapidement.



Dans un logiciel, une part de gâteau serait une partie livrée du logiciel afin que l’utilisateur puisse l’évaluer et voir l’avancement du travail. Chaque partie livrée permet de mieux gérer les risques et d’être plus confiant.

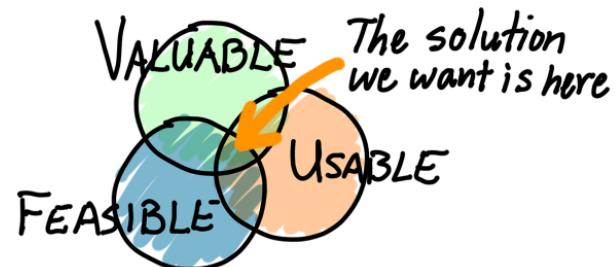
## 4.1 Epic, stories : le découpage



La taille de la story vue par les différents acteurs

- Business : aide l'entreprise à atteindre ses objectifs.
- Utilisateur : accomplit un besoin
- Développeur : prend quelques jours à dev et test

**Affirmation 10.** La conversion est l'une des meilleures choses pour découper de grosses stories



Les Epic sont des grosses stories qui ont la bonne taille pour la partie *business* ou *utilisateur* mais inadaptée pour l'équipe de *développement*. Une story doit apporter de la valeur, être utilisable et faisable

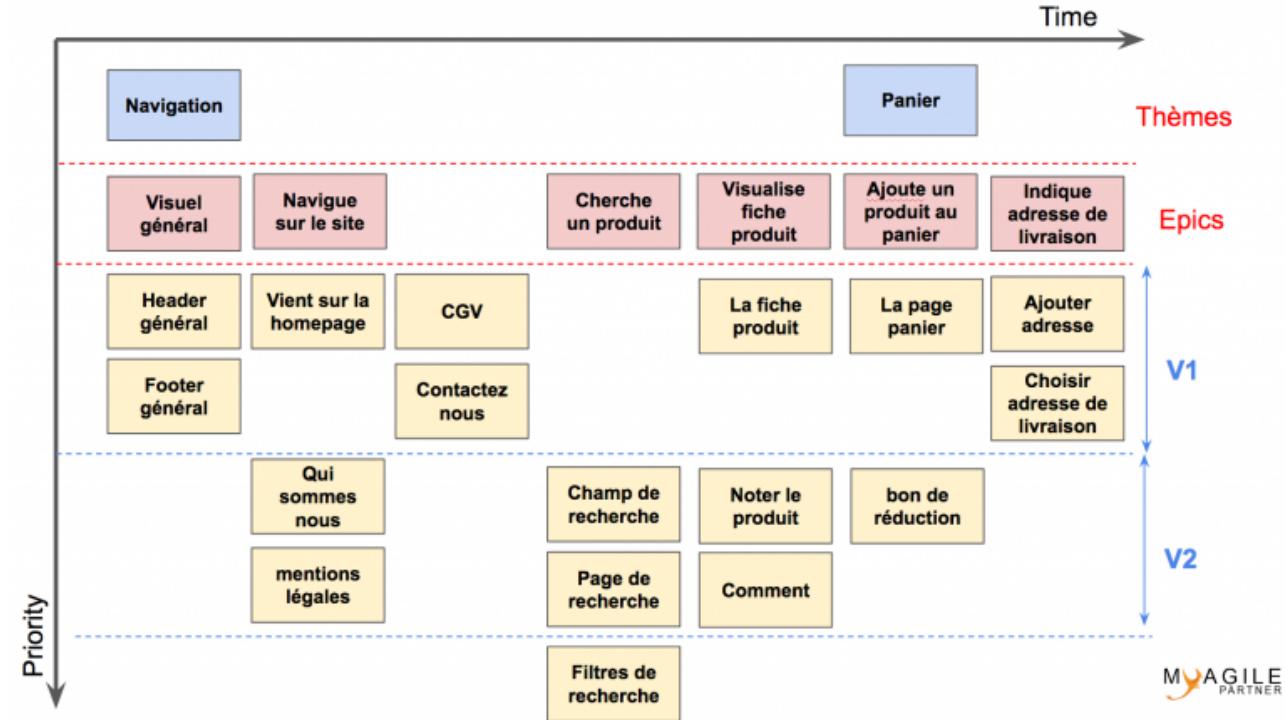
## 4.2 Organiser les stories



Le *theme* permet de regrouper un ensemble de stories qui sont fortement liées. On peut utiliser un thème pour rassembler un ensemble d'histoires qui sont nécessaires pour une prochaine version, qui font partie de la même fonctionnalité, qui sont pertinentes pour un type d'utilisateur particulier, ou qui sont liées d'une autre manière.

### 4.3 Exemples

**Definition.** L'idée de ce graphique sera de représenter les besoins utilisateurs d'un nouveau produit. Nous allons définir les usages de l'utilisateur cible et les représenter selon les priorités que nous désirons appliquer.



Afin de compléter le thème *navigation* nous devons réaliser les épics *visuel général*, *naviguer sur le site*, *chercher un produit* et *visualise fiche produit*. Chacune de ces épics se décompose en user-stories qui seront réalisées sur deux temps : la version 1 puis la version 2