



Gestion de Projets Agile

Vision, feature, user story et
story map

Tianxiao.Liu@cyu.fr

CY Cergy Paris Université

Master IISC pro 2^e Année

Plan

- Vision et features du projet
- User story (ou story)
- Décomposer les features et stories
- Techniques d'estimation agile
 - Méthode par story points
 - Démarche WBD (Wide Band Delphi)
- Planification avec story map

Vision du projet

- Définition
 - La vision est l'expression d'une volonté de développer un excellent produit ou service.
- Exemple
 - *Faire un site de rencontre pour animaux car les gens dépensent beaucoup d'argent pour leurs petites bestioles. Nous pourrions ainsi vendre de l'accompagnement, des produits et des publicités.*

Vision du projet

- Exemple formulé

Pour	Les propriétaires d'animaux domestiques
Qui	Aiment faire partager l'amour pour leurs animaux
Produit	Un site de rencontres en ligne
Qui permet	De faciliter les rencontres entre les animaux et entre leurs maîtres
A la différence de	Des rencontres au hasard
Notre produit	Apporte la possibilité de choisir ses rencontres

Feature (fonctionnalité essentielle)

- Définition
 - Une **feature** est
 - 1) un service fourni par l'équipe
 - 2) observable de l'extérieur
 - 3) qui contribue à un impact
 - 4) dont la description se situe à un niveau tel que toutes les parties prenantes comprennent facilement ce dont il s'agit.
- Notion d'impact
 - Cela ressemble à une expression de besoin mais dépasse le niveau du besoin d'utilisateur

User story

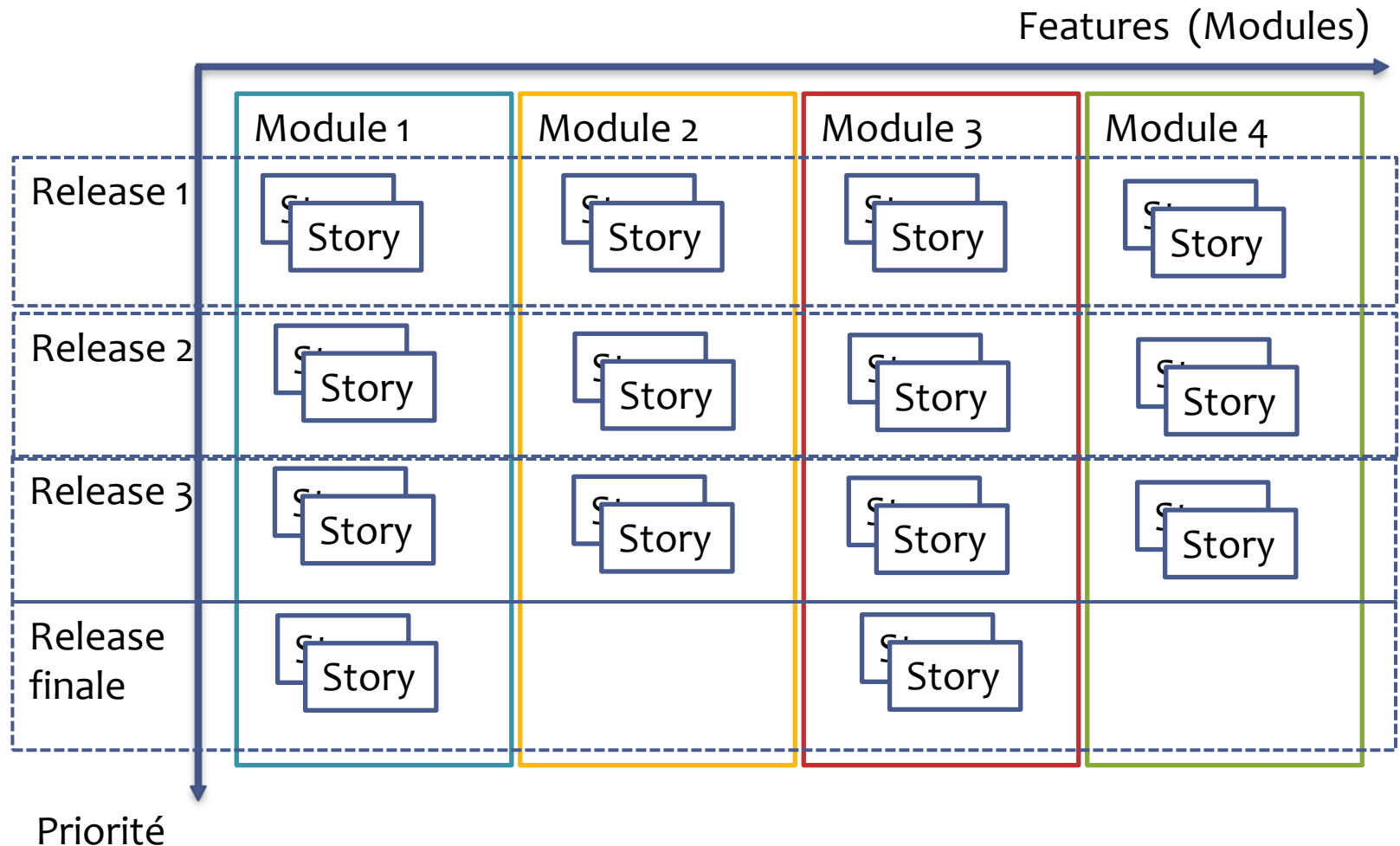
- Rappel de définition
 - *Dans une approche de gestion de projet agile, une **user story** y est présentée comme un rappel pour une conversation, lancée dans le but de faciliter la planification.*
- Objectif :
 - La user story est un morceau fonctionnel qui apporte de la valeur et qui pourra être développé en un sprint

Décomposer features → stories

- Approche moderne : **Story Map**
 - Une carte à deux dimensions
 - Dimension 1 : Features (ou modules) du projet
 - Dimension 2 : Priorité des stories
 - Squelette de l'application : une liste de features (modules)
 - Objectif : *obtenir* une première ligne en dessous du squelette → **la première release**

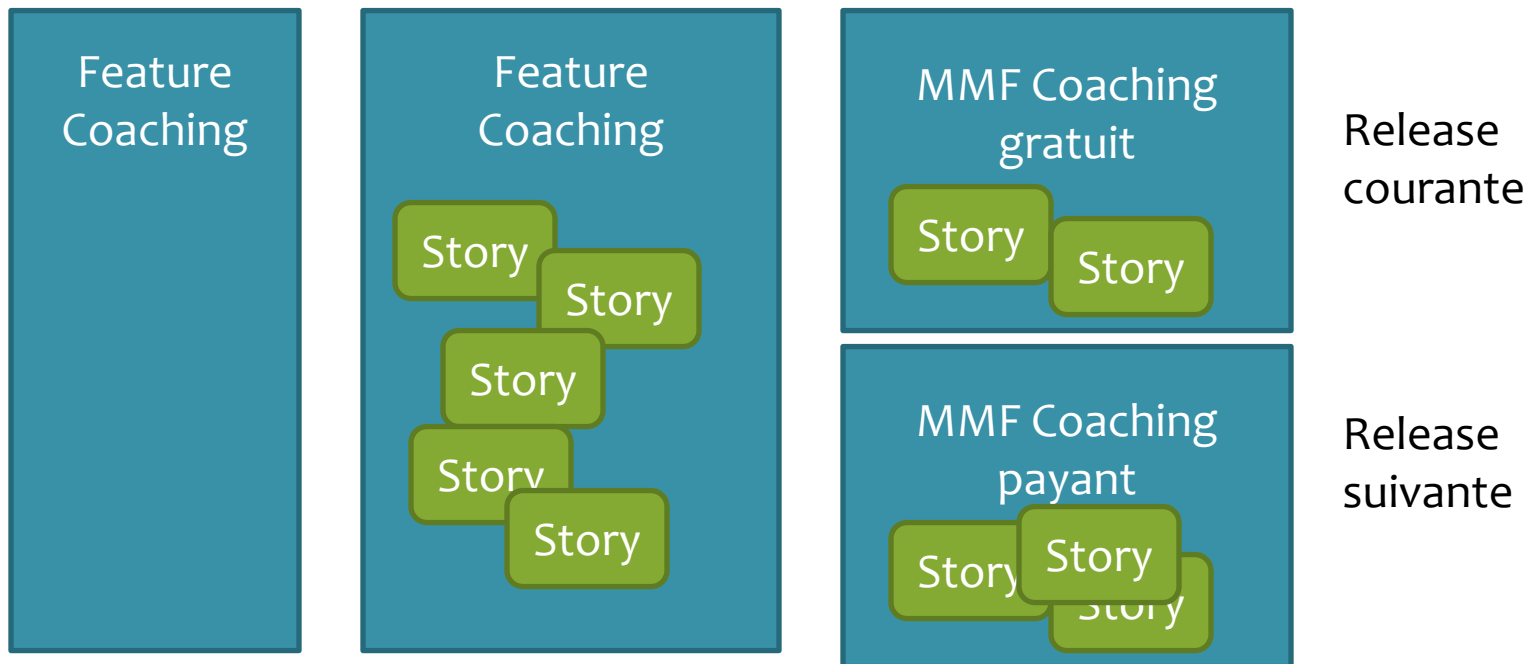
Décomposer features → stories

- Approche moderne : **Story Map**



Décomposer features → stories

- MMF : la feature minimale déployable
- Un exemple avec MMF



Plan type d'une story

- Formulation
 - As a <type of user>, I want <some goals> so that <some reason>
 - En tant que <acteur>, je veux <un but> afin de <une justification>
- Exemple
 - *En tant que membre du site, je veux demander un conseil à un éleveur afin de savoir quoi faire avec mon toutou qui aboie tout le temps.*

Plan type d'une story

- Format liste avec exemple
 - **Acteur (qui ?)** : *organisateur*
 - **Intention fonctionnelle (quoi ?)** : *connaître le nombre d'inscrits*
 - **Justification (quel impact ?)** : *choisir une salle avec la bonne capacité*
- Conseil
 - Quand on débute le projet, cette manière de formuler une user story est très utile pour éclairer sa valeur apportée

Techniques de décomposition

- Technique 1 : **variation sur les données**
- Exemple de story à décomposer
 - En tant que maître sur le site, j'ajoute un animal
 - Décomposée en 3 stories :
 - En tant que maître, j'ajoute un chien
 - En tant que maître, j'ajoute un chat (différent du chien, on ne le promène pas)
 - En tant que maître, j'ajoute un caméléon (pas de photo ! 😊)

Techniques de décomposition

- Technique 2 : **selon les règles de gestion**
- Exemple de story à décomposer
 - On peut rechercher les ancêtres d'un chien à pedigree, mais c'est réservé aux maîtres ayant acheté pour plus de 100 euros de croquettes, qui sont membres du site depuis plus d'un an.
 - Décomposée en 2 stories
 - En tant que maître, je cherche les ancêtres de mon chien (sans vérifier les droits).
 - En tant que maître, je n'accède à la recherche des ancêtres que si j'ai les droits pour cela.

Techniques de décomposition

- Technique 3 : **selon exigence fonctionnelle ou non fonctionnelle**
- Exemple de story à décomposer
 - Une story de recherche nécessitant une performance plausible
 - Décomposée en 2 stories
 - Une première qui a pour but de fournir les résultats
 - Une autre qui a pour but d'optimiser la recherche pour afficher les résultats en moins de deux secondes

Planification

- Activités de planification
 - Souvent très difficiles
 - Estimation initiale \neq Engagement
 - *Plan de route* = prévoir le processus, les étapes, les ressources, le délai et le budget, en se basant sur les besoins recueillis
 - Techniques d'estimation \neq sciences exactes
 - Aucune recette miracle fiable à 100%...

La stratégie de planification

- Propositions / remarques
 - Rien ne se passe jamais exactement comme on l'a prévu...
 - **La certitude de l'incertitude**
 - Accepter l'incertitude, non la combattre...
 - La capacité de s'adapter aux spécificités de chaque projet (équipe, client) est la qualité la plus honorable du chef de projet
 - Plus d'expériences = Meilleure spéculation

Techniques d'estimation agiles

- Méthode "Story points" (1)
 - Principe : comparaison de deux stories ou le poids d'une story dans l'ensemble → **relatif**
 -
 - Etape 1 : trouver la story la plus petite → poids de 1
 - Etape 2, comparer les autres stories avec cette première : linéaire ou **suite de Fibonacci**
 - **$F_0 = 0, F_1 = 1, F_2 = 1, F_3 = 2, F_4 = 3, F_5 = 5 \dots$**

Techniques d'estimation agiles

- Méthode "Story points" (2)
 - Etape 3 : convertir le nombre de points abstrait en jour
 - **Vélocité de l'équipe** : la somme des story points qu'une équipe est capable de développer durant une itération
 - A l'issue de chaque itération (surtout la première), on doit **re-calibrer** cette vélocité qui était hypothétique

Techniques d'estimation agiles

- Complément important : WBD
 - Principe de "la sagesse des foules"
 - Trois experts se réunissent pour estimer tour par tour les charges
 - Après 2 ou 3 tours, la charge est calculée avec la formule :

$$(P + (4 \times I) + O) / 6$$

où P = estimation pessimiste

I = estimation intermédiaire

O = estimation optimiste

Planification agile

- Vision globale du projet
 - Définir un **product backlog** (PB) initial
 - Backlog : l'ensemble de tâches qui devraient être accomplies : cahier des charges
 - A ce stade d'avancement, les fonctionnalités ne sont pas homogènes en granularité
 - Evaluer très grossièrement l'importance/la taille des différentes fonctionnalités
 - Positionner-les dans différentes releases

Travail sur la planification

- Dresser l'état des lieux
 - Votre vision du projet est-elle toujours claire ?
 - Ce qui est déjà fait
 - Ce qui est en cours et ce qui est à faire (futur proche)
- Document à envoyer pour 16H15 (à intégrer plus tard dans le chapitre GP du rapport)
 - Déterminer un plan de différentes releases
 - 1 release intermédiaire : début mars
 - 1 release intermédiaire : pré-soutenance en avril
 - 1 release intermédiaire : mai
 - 1 release finale : juin
 - **Ce qui est à réaliser pour chaque release (story map)**
 - Auto-évaluation en ce moment (dans le mail)
 - Relation avec le tuteur technique : discussions, techniques, points d'avancement, etc.