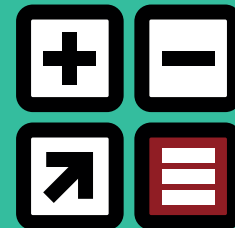


SOAT



SPLIT POKER

Deviens un Maître
du découpage
de User Story



SPLIT POKER



Crédits

Un jeu de Olivier Le Lan
Graphisme et fabrication par SOAT

Impression

Imprimer en A4 recto-verso bords courts,
et en couleur

le reste de la complexité fonctionnelle s'appuyant sur ce mécanisme .
Prenez l'exemple de l'ajout du paiement par CB sur un site web. Dans un premier temps, seul un type de CB sera géré, avec tout le mécanisme adéquat. Puis, d'autres User Stories vont enrichir cette feature, avec d'autres types de CB.



4. SIMPLE VS. COMPLEXE

Une fonctionnalité perçue comme "basique" par les clients l'est en fait rarement. On peut généralement identifier un noyau fonctionnel plus simple au sein de cette fonctionnalité, que l'on pourra alors décrire dans une User Story dédiée.

Les User Stories suivantes vont embarquer les autres cas d'usages et la complexité restante.



5. VARIATION DES DONNÉES

Un même traitement peut s'appliquer sur des données issues de sources différentes, dans des formats hétérogènes. L'exemple typique ? L'importation de données ! On peut séparer les types de formats à gérer en autant d'User Stories, c'est facile à faire et ça réduit la complexité de vos User Stories.



6. DESIGN EN 2 TEMPS

Parfois, on se lance corps et âme dans la réalisation d'une interface ultra design avec des UX designers débordant de créativité, sans avoir validé l'intérêt de la soi-disant " killer feature " par un prototype simple. Le risque que la fonctionnalité ne soit finalement pas retenue est réel, aussi pour démontrer au plus tôt la valeur apportée, il peut être intéressant de découper la User Story au niveau de l'interface : avant de montrer ce superbe nouveau widget de calendrier dernier cri, un simple champ texte peut suffire à démontrer la feature se cachant derrière.

SPLIT POKER

Un jeu de cartes pour les Product Owners et les équipes agiles !

- Comment utiliser ce jeu ?
- Présentation des patterns

SOAT



12. VARIATION DE PLATEFORME

Web, Mobile App, Desktop, etc. La plupart des applications clientes sont déclinées sur plusieurs plateformes. Mettre le focus sur une plateforme en particulier pour une feature permet de réduire la complexité d'une User Story.



BONUS 13. ETUDE ET IMPLÉMENTATION

Il se peut que le découpage soit particulièrement délicat lorsque le besoin est flou ou très complexe. Dans ce cas, il est nécessaire de se focaliser sur un scope fonctionnel plus réduit et de reprendre les patterns

de découpage sur ce sous-ensemble. Si cette étape n'est pas possible, la "Spike Story" vient à la rescousse ! Il est alors temps de formuler des hypothèses et de faire un minimum d'expérimentations pour y répondre. Dès lors, l'exercice de découpage peut reprendre, en utilisant les différents patterns précédents.



7. VARIATION D'INTERFACE

Certaines fonctionnalités impactent un nombre conséquent d'écrans, il peut alors être pertinent de découper l'implémentation par groupe d'écrans. Ce découpage peut permettre de paralléliser l'implémentation de ces nouvelles User Stories, et surtout, il permet de diminuer le nombre de tests nécessaires à réaliser pour valider la bonne implémentation de chacune.

COMMENT UTILISER CE JEU ?

Ce jeu s'utilise généralement lors des séances d'affinage de backlog. Il permet ainsi de sensibiliser les participants aux différents patterns de découpage fonctionnels, tout en profitant de l'intelligence collective des participants.

Je vous donne 2 façons d'animer ce jeu. Celui-ci étant très libre, vous pourrez tout à fait inventer vos propres règles !

Pour ces deux animations, nous avons besoin de :

- Un backlog des items à découper
- Les cartes de pattern réparties entre les participants

ANIMATION 1 : LA COURSE AU PATTERN

Une fois la distribution effectuée, on déroule le backlog : pour chaque User Story supérieure à un seuil de complexité défini (allez, par exemple > 5 points) chacun va proposer un découpage par rapport aux cartes qu'il a en main. Le plus rapide prend la main ! On découpe en direct la User Story. On recommence jusqu'à ce que sa complexité soit réduite à un niveau acceptable par les membres de l'équipe.

ANIMATION 2 : LE TOUR DES QUESTIONS

Pour chaque User Story considérée trop complexe, chaque participant va tour à tour lire les questions de sa ou ses cartes.

Si un pattern est applicable, "go !" : on découpe alors immédiatement (sans chercher à tester toutes les options de pattern).

PRÉSENTATION DES PATTERNS



1. ÉTAPE DE WORKFLOW

Si la User Story décrit un workflow, elle peut être découpée en un workflow simple (deux étapes : la première et la dernière) puis enrichie avec d'autres User Stories adressant les autres étapes du workflow global.



2. VARIATION DES RÈGLES MÉTIER

Si la User Story décrit plusieurs règles métiers, notamment dans les critères d'acceptation, on peut tenter d'alléger la liste des critères en créant des User Stories dédiées pour certaines règles.



3. EFFORT PRINCIPAL

Parfois, une User Story nécessite d'implémenter un mécanisme interne complexe et/ou nouveau. Elle peut être découpée en implémentant le mécanisme avec une fonctionnalité fonctionnellement basique, et enrichie par de futures User Stories emportant



8. PERFORMANCE EN 2 TEMPS

Les standards actuels en matière de performance vont de plus en plus haut, en particulier au niveau du temps de réponse de l'interface, plus encore que sur le temps de traitement derrière. On parle de performance perçue. Ce critère de performance peut arriver dans un second temps, après avoir montré la fonctionnalité sous-jacente. "Charger une liste d'auto-complétion" (fonction, pas de performance particulière) puis "réduire le temps de chargement en dessous de 100ms" (fonction optimisée) est un cas typique de découpage selon les performances attendues.



9. OPÉRATIONS

Lorsqu'une User Story décrit de multiples opérations, tel dans le cadre d'une implémentation de CRUD*, chaque opération peut être décrite dans une User Story dédiée.

*CRUD : opérations de **C**reate **R**ead **U**pdate **D**eleter d'une donnée



10. CIBLE UTILISATEURS

Quand une User Story implique plusieurs rôles, une possibilité intéressante à expérimenter est de séparer les spécificités d'implémentation liées à chaque rôle dans une User Story dédiée à celui-ci.



11. HAPPY / UNHAPPY

Un découpage que j'aime beaucoup car il peut s'appliquer très souvent et est assez efficace. Il s'agit de séparer les scénarios décrivant un comportement "idéal" de ceux décrivant des situations anormales, souvent liées à des erreurs de saisie utilisateur ou des risques de sécurité (je pense notamment aux "Evil User Stories"). Une User Story pour le scénario "tout va bien", et une User Story par scénario avec erreurs, et hop ! Simple et efficace.

#1

**ÉTAPE DE
WORKFLOW**

La story décrit-elle un workflow ?



Pouvez-vous découper la story en implémentant le début et la fin du workflow, puis enrichir celui-ci ultérieurement ?

#2

**VARIATION
DES RÈGLES
MÉTIER**

La story décrit-elle plusieurs règles fonctionnelles ?



Pouvez-vous découper la story en incluant un sous-ensemble de règles fonctionnelles à implémenter en priorité ?

#3

**EFFORT
PRINCIPAL**

Le besoin fonctionnel de la story nécessite-t-il d'implémenter une infrastructure / un mécanisme nouveau ?



Pouvez-vous découper la story en implémentant d'abord l'infrastructure via une fonctionnalité simple ?

#4

**SIMPLE /
COMPLEXE**

La story contient-elle un noyau fonctionnel simple apportant un maximum de valeur ?



Pouvez-vous découper la story en implémentant d'abord ce noyau fonctionnel, puis en l'enrichissant plus tard ?

#5

**VARIATION
DES
DONNÉES**

Le comportement décrit est-il le même pour différents types de données ?



Pouvez-vous découper la story pour traiter d'abord un seul type de données, et étendre ensuite à d'autres types ?

#6

**DESIGN EN
2 TEMPS**

La story comporte-t-elle une interface complexe ?



Y'a-t-il une version plus simple par laquelle commencer ?

#7

**VARIATION
D'INTERFACES**

La story implique-t-elle le traitement d'une donnée identique via plusieurs interfaces ?



Pouvez-vous découper la story pour ne traiter qu'un type d'interface par story ?

#8

**PERFORMANCE
EN 2 TEMPS**

La complexité de la story réside-t-elle dans des exigences non fonctionnelles comme la performance ?



Pouvez-vous découper la story pour satisfaire d'abord le côté fonctionnel et l'enrichir ensuite avec des exigences telles que la performance ?

#9 OPÉRATIONS (CRUD)



La story inclut-elle de multiples opérations ?
Parle-t-elle de gestion ou configuration ?



Pouvez-vous séparer chaque opération en une story dédiée ?

#10 CIBLE UTILISATEURS



La story implique-t-elle plusieurs rôles ?



Pouvez-vous découper la story pour ne gérer qu'un rôle par story ?

#11 HAPPY / UNHAPPY



La story implique-t-elle la gestion d'erreurs ou de risques ?



Pouvez-vous découper la story pour ne gérer que le scénario idéal, et l'enrichir ensuite par des scénarios d'erreurs ou de mauvais comportements ?

#12 VARIATION DE PLATEFORME



La story décrit-elle le même comportement sur plusieurs plateformes ?
(site web, mobile, etc.)



Pouvez-vous découper la story pour ne gérer qu'une plateforme par story ?

#13 ÉTUDE ET IMPLÉMENTATION



La story comporte-t-elle plusieurs incertitudes ?



Pouvez-vous définir les 1 à 3 questions qui vous freinent le plus ?

UN MOT SUR L'AUTEUR

Olivier Le Lan est Coach Agile et grand amateur de jeux. Il accompagne les organisations pour créer des écosystèmes de travail apprenants, performants, bienveillants. Avec des pratiques ludiques et une dose de fun !

SPLIT POKER

Un jeu de cartes pour les Product Owners et les équipes agiles !

- Pourquoi ce jeu ?
- Un mot sur l'auteur

SOAT

**SPLIT
POKER**



PATTERN

SCAT

**SPLIT
POKER**



PATTERN

SCAT

**SPLIT
POKER**



PATTERN

SCAT

**SPLIT
POKER**



PATTERN

SCAT

**SPLIT
POKER**



PATTERN

SCAT

**SPLIT
POKER**



PATTERN

SCAT

**SPLIT
POKER**



PATTERN

SCAT

**SPLIT
POKER**



PATTERN

SCAT



POURQUOI CE JEU ?

Le découpage fonctionnel est tout un art, une compétence que les Product Owners et autres rôles fonctionnels se doivent de maîtriser pleinement pour réduire la complexité du projet et ainsi réussir à tenir les grandes promesses de l'agilité :

- Livrer plus vite les fonctionnalités clés
- Casser l'effet tunnel
- Améliorer la qualité
- Diminuer les risques

Dans le cas d'une équipe de réalisation, cela passe par un découpage pertinent de l'appliquatif à produire. Aujourd'hui l'une des grandes problématiques pour tous ces rôles fonctionnels est l'efficacité du **découpage fonctionnel**.

Par chance, le découpage est une compétence et donc, peut se travailler. Oui, mais comment apprendre efficacement et s'exercer en minimisant le côté parfois rébarbatif ? Une solution : la gamification !

C'est dans cette optique qu'Olivier Le Lan a créé le Split Poker, un petit jeu de cartes qui ludifie les sessions de découpage fonctionnelles et rend celles-ci plus efficaces. Pour créer ce jeu, il s'est largement inspiré des travaux de Richard Lawrence et de son célèbre poster "How To Split a User Story" (cf. <https://agileforall.com/how-to-split-a-user-story/>).

En tant que voyageur
je veux payer mon
billet d'avion pour
booker mon voyage.



- ☐ Je choisis ma place
- ☐ Je choisis mon repas
- ☐ Je choisis mon assurance
- ☐ Je règle mon billet

ESSAYER #1

En tant que voyageur au
budget serré et flexible
sur mes dates je veux
réaliser une recherche
de vol avec dates
flexibles pour trouver
le meilleur prix.



- ☐ Dates flexibles à 3 jours
- ☐ Week-end sur 2 prochains mois
- ☐ 1 semaine calendaire sur une période donnée

ESSAYER #2

En tant que voyageur
je veux une interface
sécurisée pour régler
mon billet d'avion.



- ☐ Je peux payer en CB VISA
- ☐ Je peux payer avec les 3 types de carte de crédit (VISA, MC, AMEX)

ESSAYER #3

En tant que voyageur
je veux paramétrer ma
recherche pour trouver
le trajet le moins
épuisant.



- ☐ Nombre d'escales maximum
- ☐ Limiter à certains aéroports
- ☐ Temps d'escale maximum

ESSAYER #4

En tant que voyageur
je veux trouver le
meilleur trajet pour
joindre 2 destinations.



- ☐ Liste des vols
- ☐ Liste des trains
- ☐ Liste des bus
- ☐ Liste des bateaux

ESSAYER #5

En tant que voyageur
je veux renseigner mes
dates de voyage pour
borner ma recherche.



- ☐ Saisie textuelle des dates
- ☐ Saisie par un calendrier ergonomique
- ☐ En langage naturel

ESSAYER #6

En tant que voyageur
je veux accéder à ma
réservation pour me
rappeler de ses détails.



- ☐ Mail de compte-rendu en fin de réservation
- ☐ Réservation dans la section « mon compte »
- ☐ SMS la veille du départ

ESSAYER #7

En tant que voyageur
je veux disposer d'une
vraie UX pour faire ma
recherche de vol.



- ☐ Une animation me signifie que la recherche est en cours
- ☐ La recherche se fait en moins de 5 secondes

ESSAYER #8

En tant qu'utilisateur enregistré je veux gérer mon compte pour contrôler mes données.



- ☐ Modifier mes informations
- ☐ Supprimer mon historique
- ☐ Désactiver mon compte
- ☐ Supprimer mon compte et ses données liées

ESSAYER

#9

En tant que voyageur je veux trouver une escape pour partir en week-end.



- ☐ En dernière minute au soleil avec des potes
- ☐ En couple pour fêter notre anniversaire de rencontre
- ☐ Rentrer au pays voir la famille

ESSAYER

#10

En tant que voyageur je veux me connecter pour consulter mes réservations.



- ☐ Connection sécurisée
- ☐ Mot de passe oublié
- ☐ Tentative de « brute force » sur le compte d'un client

ESSAYER

#11

En tant que voyageur je veux utiliser différents médias suivant mon environnement pour vérifier ma réservation.



- ☐ Sur mon poste fixe chez moi
- ☐ Sur mon smartphone
- ☐ Sur une impression papier

ESSAYER

#12

En tant que possesseur d'une enceinte connectée je veux utiliser ce média pour améliorer mon UX.



- ☐ Prise en compte de mes préférences dans les questions posées
- ☐ Avis sur la pertinence du lieu de destination vis-à-vis de la météo

ESSAYER

#13

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT

**SPLIT
POKER**

EXEMPLE

SOAT