

R5.04 – Cycle de vie d'un projet informatique

TD 1 : planifier un sprint

Halim Djerroud

révision 1.0

Objectifs et organisation

Ce TD met en pratique le **cours 1** : écrire des user stories, les estimer, les découper en tâches et suivre leur avancement. Tout se fait **sur papier**, sans machine : l'objectif est de travailler la méthode, pas l'outil. L'outil viendra au TP, dans votre forge.

Le fil conducteur est la SAÉ 5.01/5.02 « La Carotte Électronique » : un porte-monnaie électronique sur carte à puce pour l'IUT. On y trouve les applications **Lubiana** (administration des cartes), **Rodelika** (gestion des bonus), **Berlicum** (borne de recharge) et **Lunar White** (distributeur de boissons), ainsi que la base de données **Purple Dragon**.

Organisation

- Vous travaillez en **équipe de SAÉ** (2 ou 3 étudiants) : les exercices 3 et 4 n'ont aucun intérêt seul.
- Les exercices 1 à 4 portent sur un énoncé commun : ils seront **corrigés au tableau**.
- L'exercice 5 porte sur **vosre** SAÉ : ce que vous y produisez sera saisi dans la forge au TP.
- Prévoyez des **ciseaux** pour l'annexe (cartes de planning poker), ou utilisez une application de planning poker sur téléphone.

Déroulé indicatif (2 h)

Ex.	Contenu	Durée
1	Du besoin à la user story : INVEST et epics	25 min
2	Critères d'acceptation et découpage en tâches	25 min
3	Planning poker et vélocité	35 min
4	Kanban, limite de WIP et Definition of Done	25 min
5	Le backlog de votre SAÉ (préparation du TP)	10 min

À rendre en fin de séance

Une feuille par équipe portant les réponses de l'exercice 5 : six user stories estimées, votre Definition of Done, et la sélection du sprint 1. **Apportez-la au TP** : vous la saisirez dans votre forge.

Exercice 1 : du besoin à la user story (25 min)

Rappel du cours. Une user story s'écrit **En tant que** <rôle>, **je veux** <action>, **afin de** <bénéfice>. Elle doit respecter les critères **INVEST** : Indépendante, Négociable, apporte de la Valeur à l'utilisateur, Estimable, Small (tient dans un sprint), Testable.

Lors du premier rendez-vous avec le client, votre équipe a noté les six formulations suivantes, telles qu'elles ont été prononcées :

- (a) « *Faire la base de données Purple Dragon.* »
- (b) « *En tant qu'agent administratif, je veux gérer les cartes des étudiants afin de faire mon travail.* »
- (c) « *Le système doit débiter 20 centimes sur la carte lors de l'achat d'une boisson.* »
- (d) « *En tant qu'étudiant, je veux que la borne de recharge soit rapide.* »
- (e) « *En tant qu'étudiant, je veux consulter mon solde, recharger ma carte et voir l'historique de mes achats afin de gérer mon argent.* »
- (f) « *En tant que développeur, je veux utiliser une bibliothèque Python pour les APDU afin de coder plus vite.* »

1. Pour chaque formulation, indiquez le ou les critères **INVEST** qui ne sont pas respectés, et expliquez en une phrase pourquoi.
2. Réécrivez chaque formulation au format du cours. Attention : certaines donnent **plusieurs** stories, et certaines n'ont **pas leur place** dans le backlog en tant que story – dites alors où elles doivent aller.
3. Parmi les formulations de départ, laquelle est une **epic** ? Découpez-la en au moins trois stories indépendantes.
4. Une story de votre backlog est écrite ainsi : « *En tant qu'étudiant, je veux acheter une boisson afin de ne pas avoir besoin de monnaie.* » Le Product Owner vous annonce que le prix de la boisson passera de 20 à 25 centimes. Faut-il réécrire la story ? Justifiez avec le **N** d'INVEST.

Exercice 2 : critères d'acceptation et découpage en tâches (25 min)

Rappel du cours. Un **critère d'acceptation** dit *ce que* le produit doit faire, du point de vue de l'utilisateur, et doit être **vérifiable** : on doit pouvoir écrire un test qui répond oui ou non. Une **tâche** dit *comment* l'équipe s'y prend : c'est une action technique, qui devient une carte du Kanban.

Le Product Owner retient pour le sprint la story suivante (application Rodelika) :

S – *En tant qu'agent administratif, je veux attribuer un bonus à un étudiant afin de récompenser sa participation à la vie de l'IUT.*

1. Classez chacun des huit énoncés ci-dessous : **critère d'acceptation**, **tâche technique**, ou **ni l'un ni l'autre**. Justifiez les cas litigieux.
 - (a) Le bonus attribué apparaît dans la liste des opérations de l'étudiant.
 - (b) Ajouter une table **operation** dans la base Purple Dragon.
 - (c) L'interface de saisie doit être ergonomique.
 - (d) Un montant négatif est refusé, avec un message d'erreur explicite.
 - (e) Écrire les tests unitaires de la fonction `attribuer_bonus()`.
 - (f) Le code est relu par un autre membre de l'équipe avant d'être fusionné.
 - (g) Le bonus est enregistré même si l'étudiant n'a pas sa carte sur lui.
 - (h) Le montant du bonus est compris entre 1 et 50 euros.
2. Rédigez **trois critères d'acceptation vérifiables supplémentaires** pour la story **S**. Pour chacun, dites en une ligne quel test permettrait de le vérifier.

3. Découpez la story **S** en au moins **cinq tâches techniques**. Indiquez celles que deux membres de l'équipe peuvent réaliser **en parallèle**, et ce dont l'équipe doit se mettre d'accord **avant** pour que ce soit possible.
4. Le développeur qui prend la tâche « écran de saisie » annonce qu'il en a pour quatre jours, sur un sprint de deux semaines. Est-ce acceptable ? Que peut faire l'équipe ?

Exercice 3 : planning poker et vélocité (35 min)

Rappel du cours. Au planning poker on estime la **complexité relative** des stories, en **points**, jamais en heures. Chacun choisit sa carte en secret, tout le monde révèle en même temps, les estimations extrêmes s'expliquent, puis on revote jusqu'au consensus. La **vélocité** est le nombre de points terminés en un sprint.

3.1 – Jouez (15 min)

Découpez les cartes de l'annexe, une série par personne. La story **A** sert d'**étalon** : elle vaut **2 points**. Estimez les autres **par rapport à elle**, en jouant réellement : carte en secret, révélation simultanée, discussion, nouveau vote.

#	User story	Vote 1	Vote 2	Vote 3	Consensus
A	En tant qu' agent administratif , je veux consulter le solde d'une carte afin de répondre à la question d'un étudiant.	<i>étalon</i>			2
B	En tant qu' agent administratif , je veux attribuer un bonus à un étudiant afin de récompenser sa participation.				
C	En tant qu' agent administratif , je veux personnaliser une carte vierge (titulaire, numéro) afin de la remettre à un nouvel étudiant.				
D	En tant qu' étudiant , je veux acheter une boisson au distributeur avec ma carte afin de ne pas avoir besoin de monnaie.				
E	En tant qu' agent administratif , je veux bloquer une carte déclarée perdue afin que personne d'autre ne puisse l'utiliser.				
F	En tant qu' agent administratif , je veux suivre l'état de l'ensemble du parc de cartes afin de piloter leur renouvellement.				

Attention. Il n'y a pas de « bonne » valeur attendue : deux équipes peuvent estimer la même story 3 et 5 sans qu'aucune ait tort. Ce qui compte, c'est que vos estimations soient **cohérentes entre elles** et que la discussion ait eu lieu.

3.2 – Cinq situations (10 min)

1. Sur la story **B**, les trois développeurs révèlent **2**, **3** et **20**. Que doit faire l'équipe ? En quoi ce résultat est-il une bonne nouvelle ?
2. Sur la story **E**, les trois développeurs jouent la carte **?**. Que signifie cette carte, et que doit faire l'équipe avant de pouvoir estimer ?
3. Sur la story **F**, les votes sont **40**, **100** et **100**. Qu'en conclut l'équipe ?
4. Un développeur propose : « on n'a qu'à prendre la moyenne des trois cartes et passer à la story suivante, on gagnera du temps. » Que répondez-vous ?

5. Le Product Owner intervient : « *la story B, c'est 3 points, pas plus : on est en retard.* » Relevez **deux** problèmes dans cette phrase.

3.3 – Vélodité et sélection du sprint (10 min)

Sur ses trois derniers sprints, l'équipe a terminé **12**, **18** et **15** points. Le Product Owner présente le backlog, trié par ordre de priorité décroissante :

Story (par priorité)	B	D	E	G	H
Points	5	8	3	5	2

1. Quelle vélocité l'équipe doit-elle retenir ? Justifiez le calcul.
2. Quelles stories prend-elle dans le sprint ? Justifiez.
3. Un membre propose : « *prenons aussi H, elle ne fait que 2 points et il reste de la place.* » Que répondez-vous ?
4. À la fin du sprint, B et E sont terminées ; D est codée et sa pull request est ouverte, mais personne ne l'a relue. Quelle est la vélocité de ce sprint ?
5. Cette vélocité est la plus basse des quatre derniers sprints. Est-ce une mauvaise note pour l'équipe ? Qu'est-ce que ce chiffre lui dit exactement ?

Exercice 4 : Kanban, limite de WIP et Definition of Done (25 min)

Voici le tableau de l'équipe le **jour 7** d'un sprint de **10 jours**. L'équipe compte trois développeurs : *Alice*, *Karim* et *Léa*.

À faire	En cours	En revue	Terminé
Écran de saisie du bonus	Fonction attribuer_bonus <i>Alice</i>	Lecture du solde <i>Alice</i> – depuis le jour 2	Afficher la version de la carte <i>poussée directement sur main, tests non lancés</i>
Tests du blocage de carte	Table operation <i>Alice</i>	Conversion des montants <i>Karim</i> – depuis le jour 3	
Mettre à jour le README	Écran de blocage <i>Karim</i>	Menu de Lubiana <i>Léa</i> – depuis le jour 4	
	APDU de blocage <i>Karim</i>		
	Recherche d'un étudiant <i>Léa</i>		

1. Citez **trois problèmes** que ce tableau révèle. Pour chacun, dites ce qu'il annonce pour la fin du sprint.
2. Le lendemain matin, au point quotidien, dans quel **ordre** l'équipe doit-elle agir ? Justifiez.
3. L'équipe décide de **limiter le travail en cours à une carte par personne**.
 - (a) Qu'est-ce que cela change concrètement dans la journée d'un développeur ?
 - (b) Karim objecte : « *si je n'ai le droit qu'à une carte, je serai bloqué dès que j'attends une revue.* » Que lui répondez-vous ?
4. Rédigez la **Definition of Done** de votre équipe : au moins cinq critères, sous forme de cases à cocher.
5. Au regard de cette DoD, quelle carte du tableau se trouve dans la **mauvaise colonne** ? Où doit-elle retourner, et pourquoi est-ce important au-delà du cas particulier ?

Exercice 5 : le backlog de votre SAÉ (10 min, à terminer avant le TP)

Cette fois, vous travaillez sur **votre** projet. Ce que vous produisez ici sera saisi dans votre forge au TP : les stories deviendront des **tickets**, les points des **étiquettes**, et la Definition of Done ira dans le `README.md`.

1. Relisez le sujet de la SAÉ et rédigez **six user stories**, au format du cours. Vérifiez chacune avec INVEST et donnez-lui au moins **deux critères d'acceptation**.
2. Triez-les par **priorité**, du point de vue du client. En cas de désaccord dans l'équipe, notez la question à poser à l'enseignant : c'est lui qui joue le Product Owner.
3. **Estimez-les** au planning poker. Choisissez d'abord votre étalon : la plus petite story que vous savez faire, à laquelle vous donnez 1 ou 2 points.
4. Vous n'avez **aucune vélocité** : c'est votre premier sprint. Comment décidez-vous du contenu du sprint 1 ? Notez votre raisonnement – vous le comparerez au résultat réel.
5. Rédigez la **Definition of Done** de votre équipe (exercice 4, question 4), et **découpez en tâches** les stories que vous prenez dans le sprint 1.

Ce que vous apportez au TP

- Vos **six user stories**, avec leurs critères d'acceptation et leurs points.
- La **Definition of Done** de votre équipe.
- La **sélection du sprint 1** et le découpage en tâches.

La partie 3 du TP (« Créer le projet et organiser le sprint 1 ») ne sera alors qu'une **saisie** dans la forge : vous gagnerez le temps nécessaire pour la mise en place de l'intégration continue.

Annexe : cartes de planning poker

Une série par personne. Suite de Fibonacci modifiée : les écarts se creusent à mesure que les stories grossissent, parce qu'on ne sait pas estimer finement ce qui est gros. La carte ? signifie « je n'ai pas assez d'informations », et 40 ou 100 signifient « c'est une epic, il faut la découper ».

0	$\frac{1}{2}$	1	2	3	5	8	13	20	40	100	?
0	$\frac{1}{2}$	1	2	3	5	8	13	20	40	100	?
0	$\frac{1}{2}$	1	2	3	5	8	13	20	40	100	?