

M1 : Ingénierie du Logiciel – MU4IN502

SORBONNE UNIVERSITE – FACULTE DES SCIENCES

Examen Réparti 1ere partie

13 novembre 2019 (2 heures avec documents : tous SAUF ANNALES CORRIGÉES). Barème indicatif sur 20 points.

1. Questions de cours

[4 Pts]

Répondez de façon précise et concise aux questions.

Barème : VALABLE sur toutes les questions de cours : -25 à -50% si la réponse inclut la bonne idée, mais qu'elle est noyée dans des infos ou autres réponses fausses/inappropriées.

Q1.1(1 point) : Définir la nature et le rôle d'une *maquette* et d'un *prototype*.

Maquette : élément qui permet de se faire une idée (souvent visuelle) de la solution, mais qui n'interviendra pas dans le produit final (la maquette est jetable).

Par exemple : un dessin d'un écran d'IHM en GL, un modèle réduit en ingénierie classique (e.g. architecture)

Prototype : élément qui réalise une partie des fonctionnalités du produit final, par raffinements successifs le prototype à vocation à devenir une partie de la solution finale.

Par exemple : (GL) logiciel réalisant uniquement le scénario nominal, (classique) pièce automobile utilisée en Formule 1 ~> intégrée dans en série dans la génération suivante.

Barème :

50% chaque, en présentant nature (conceptuel jetable vs exécutable raffiné) et rôle (utilité au moins esquissée).

Les exemples ne sont pas obligatoires mais comptent comme une partie de l'explication

Q1.2(1 point) : Est-ce possible qu'une personne physique soit représentée par plusieurs acteurs dans un diagramme de use case ? Justifiez votre réponse en donnant un exemple ou en expliquant pourquoi ce n'est pas possible.

Oui, à des instants différents, une même personne physique peut jouer des rôles différents vis-à-vis du système.

e.g. Le responsable de magasin peut « incarner » un « caissier » à certains moments.

30% réponse OUI

30% justification sur le fait que c'est un rôle.

40% : un exemple

Q1.3 (1 point) : Sur un diagramme de classe, donnez un exemple concret de deux classes métier liées par deux associations.

Personne – Entreprise : responsable + employé de

On doit voir des

- 40% diagramme pertinent, exemple OK
- 40% noms de rôles distincts
- 20% des cardinalités

Q1.4 (1 point): La diversité est souhaitable dans une démarche de développement logiciel. Donnez deux exemples concrets de vues et expliquez leur complémentarité.

Barème :

35% *2 par vue + axe utilisé dans cette vue/public visé. Typiquement une petite description d'un des diagrammes utilisés.

Exemples :

Use Case : vue missions du système, point de vue utilisateur final, abstrait

Classes : structure statique des liens potentiels, organisation des données

Séquence : dynamique des interactions entre objets

...

30% sur la cohérence/explication de comment ces vues se touchent, et/ou discours sur le fait de ne pas tout mélanger dans un seul diagramme, la réponse montre que l'on comprend le terme « diversité » tel qu'il est employé ici. On note assez généreux ici ; forte variabilité des réponses.

2. Problème: Analyse de StoneHearth [16 Pts]

Le Jeu "StoneHearth" est un jeu à deux joueurs, où les joueurs s'affrontent en jouant chacun leur tour des cartes. La particularité du jeu par rapport aux jeux de cartes classiques (poker, belote...) est que les cartes sont toutes différentes et dotées d'effets particuliers. Chaque carte a un nom et une description qui explique informellement ses effets. Chaque carte a également une valeur d'attaque et une valeur de défense (deux entiers naturels). Les cartes sont rangées en quatre catégories de "rareté" croissante : "basique", "commune" puis "rare" puis "légendaire". Les cartes les plus puissantes sont aussi les plus rares.

Chaque joueur a donc une collection de cartes, à partir de laquelle il devra composer son "deck", c'est-à-dire choisir les 30 cartes (sans doublon) qu'il utilisera pour affronter son adversaire. Le joueur dispose d'une interface lui permettant d'éditer le nom du deck et les cartes constituant un deck, mémorisé dans un des trois emplacements dont il dispose. Il peut débloquer d'autres emplacements de decks pour 2eu ou 3\$ chaque emplacement. Un deck incomplet (moins de 30 cartes) peut être mémorisé, mais pas utilisé pour jouer une partie (il sera complété plus tard).

Pour ajouter des cartes à sa collection, le joueur doit ouvrir des "packs", contenant 5 cartes aléatoires dont au moins une "rare". Les nouveaux joueurs ont droit automatiquement à toutes les cartes "basique" ce qui leur permet de composer un deck, plus 5 packs gratuits pour se lancer. Les packs peuvent être achetés dans le jeu pour 1,39eu ou \$1.99. Les joueurs assidus se voient également attribuer un pack toutes les 10 parties jouées.

Comme les cartes issues des packs sont aléatoires, mais que les doublons sont interdits dans un deck, un joueur ayant des cartes en double peut vouloir détruire une carte ce qui lui donne des "joyaux". Les joyaux sont une monnaie dans le jeu, qui ne sert qu'à acheter des cartes, mais au choix de l'utilisateur cette fois. La seule façon d'en obtenir est de détruire des cartes de sa collection (typiquement les cartes obtenues en double, mais pas forcément). Détruire une carte commune rapporte 2 joyaux, une carte rare 5 joyaux, une carte légendaire rapporte

20 joyaux. Les cartes "basique" ne peuvent pas être détruites ni obtenues dans les packs. Pour acheter une carte commune il faut 20 joyaux, une carte rare 50 joyaux, une carte légendaire coûte 200 joyaux.

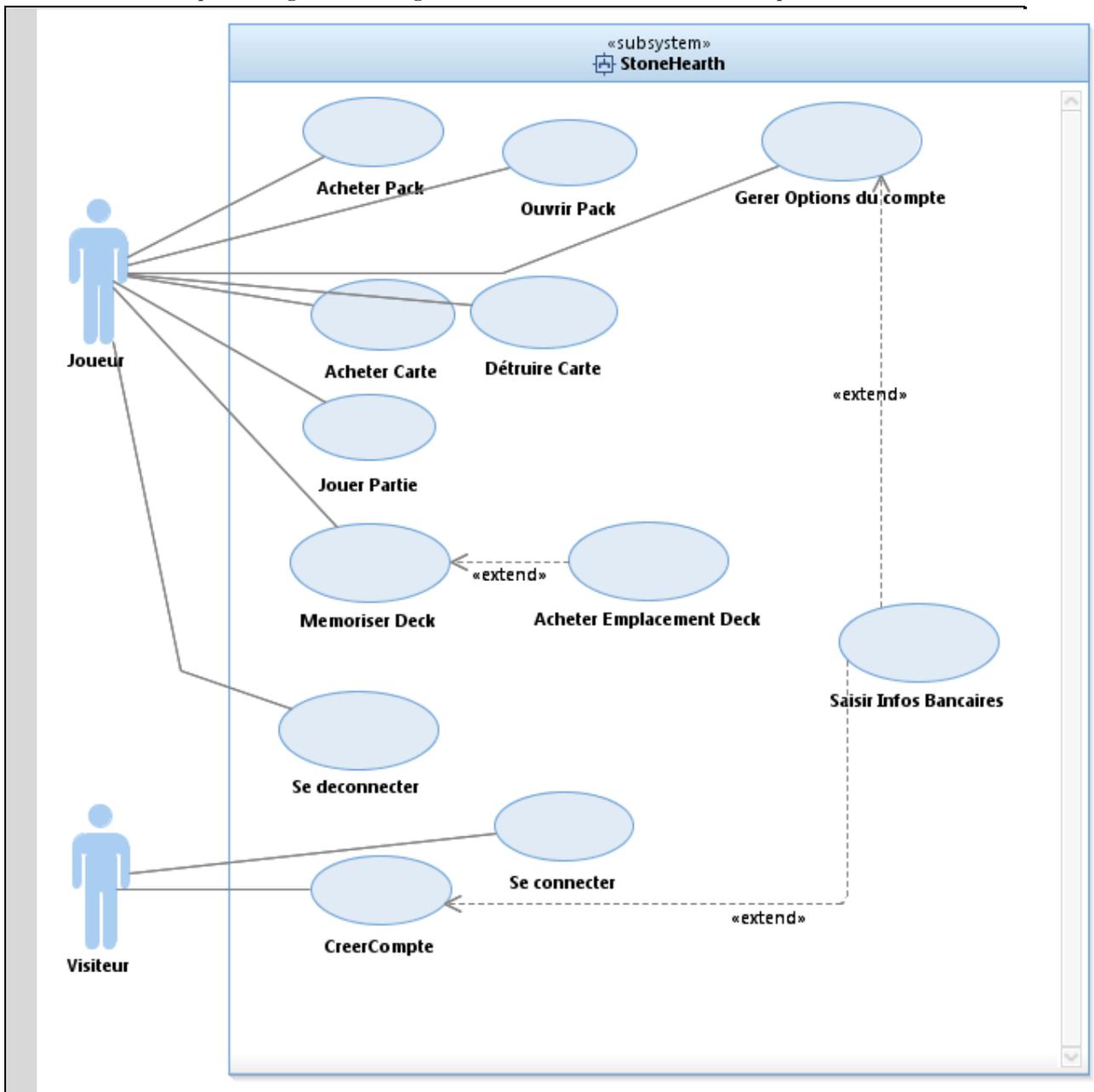
Les joueurs doivent créer un compte pour se connecter au jeu et accéder à leur collection de cartes et leurs decks mémorisés. Ils doivent y renseigner leurs coordonnées bancaires pour pouvoir acheter des packs et des emplacements, mais ce n'est pas obligatoire de le faire immédiatement. Ils peuvent accéder aux options du compte par la suite, et les saisir quand ils le souhaitent.

Les joueurs connectés peuvent alors directement décider de jouer contre un joueur de même rang. Ils choisissent un deck dans la liste des decks qu'ils ont mémorisé, puis le jeu leur trouve un adversaire de rang similaire (écart ≤ 3 rangs). Le rang des nouveaux joueurs est initialement 0. Chaque partie gagnée incrémente le rang (sauf s'il est 100), chaque partie perdue le décrémente de 1 (sauf s'il est à 0).

Les joueurs jouent alors chacun leur tour des cartes jusqu'à ce que l'issue soit décidée par la victoire d'un des joueurs; il n'y a pas de match nul possible. Dans cet énoncé on ne détaillera pas les règles du jeu lui-même, ni le déroulement de la partie.

Question 2.1 : (3,75 pts) Réalisez le diagramme de cas d'utilisation de la phase d'analyse. Vous justifierez tous vos choix, par un texte ou des annotations sur le diagramme.

Acteur : joueur



Barème :

Use case du Joueur : (10% chacun * 8) :

Acheter Pack, Ouvrir Pack

Acheter Carte, Détruire Carte (on accepte gérer cartes si on a un commentaire)

Jouer partie

Mémoriser Deck, Acheter Emplacement

Options du compte (CB...)

Gestion correcte de l'authentification : 20%

Inclut Créer Compte et Se Connecter (Déconnecter est optionnel)

10% seulement si on a lié ces use case à Joueur (donc un seul acteur)

Bonus : +5% pour le saisir Carte clairement optionnel

-10% par include ou extend injustifiable

Fautes fréquentes :

TODO

-5% Le use case « modifier dépôt » est hors système (c'est git qui gère ça), pénalité de -5% pas très méchante s'il apparaît.

-10% autres use cases hors cadre

-5% si c'est TROP détaillé (créer projet dans orga distingué de créer projet personnel, « fournir email » include depuis créer comptes (par contre « confirmer email » est ok, pas de pénalité)

-5 à -15% pour les fautes

-5 par use case mal formulé : on veut un verbe qui exprime l'action du point de vue de l'acteur.

-10% aucun commentaire/aucun texte pour accompagner le diagramme, diagramme sec

Jusqu'à -20% par héritage, include ou extend injustifiable ou autre incohérence ou mésusage d'UML.

-10% si on ne précise pas qui fait l'action dans le scenario (use case sans acteur lié)

Question 2.2 : (3,5 pts) Précisez la ou les fiches détaillée(s) (acteurs concernés, pré-conditions, post-conditions, scénario nominal, alternatives, exceptions) du (ou des) cas d'utilisation(s) correspondant aux interactions permettant de composer un « deck » et de le mémoriser dans un emplacement.

1 use case a priori : créer deck

Pré : être logé

1. Le joueur choisit de créer un deck
2. Le système affiche les emplacements, en indiquant les emplacements libres
3. Le joueur sélectionne un emplacement
4. Le sys affiche toutes les cartes du joueur, et la liste des cartes du deck (donc vide si on a choisi un emplacement vide)
5. Le joueur sélectionne une carte à ajouter au deck parmi la collection
6. Le système met à jour le contenu du deck et l'affichage : les cartes déjà dans le deck ne sont pas sélectionnables dans la collection.
7. L'utilisateur choisit d'enregistrer le deck
8. Le système invite à saisir un nom pour le deck
9. L'utilisateur saisit le nom du deck et valide
10. Le système mémorise le deck

Post : le deck nouvellement créé est disponible pour jouer.

A1 : ajout autre cartes

En SN7, l'utilisateur peut au contraire continuer tant que le deck comporte moins de 30 cartes, retour en SN5

A2 : enlever une carte

En SN5, l'utilisateur peut sélectionner une carte du deck en cours de construction,

A2.2 Le système retire la carte sélectionnée du deck.

A2.3 Retour en SN6.

A3 : emplacement occupé

En SN3, si l'utilisateur choisit un emplacement non vide, le système initialise l'affichage avec le contenu actuel du deck, retour en SN4.

Barème : sur 100% TODO

10% Précondition sur l'utilisateur est connecté pour pouvoir démarrer, 5% si on note que acteur concerné = « joueur connecté » mais pas de précondition explicite

20% affichages du système : on a clairement identifié les deux écrans importants à afficher : 1. les emplacements, libres ou occupés. 2. La collection du joueur et la composition actuelle du deck.

20% Le scénario principal permet clairement d'ajouter des cartes à un deck, 10% si vague, mal spécifié, mélangé avec retirer des cartes...

10% on voit passer le nom du deck

10% post condition : le nouveau deck est enregistré

10% modélisation propre de la boucle pour continuer à ajouter via une ALT qui reboucle sur le nominal

10% ALT pour enlever des cartes du deck (5% si ça apparaît autrement que dans un ALT)

10% ALT ou spécification propre dans le nominal du cas emplacement non vide

On peut éventuellement donner 10% pour des ALT ou exception bien spécifiées qui traitent d'autres cas pertinents. Exemples : renvoyer vers la boutique si pas d'emplacements libres, étapes supplémentaires pour notifier à l'utilisateur s'il enregistre un deck incomplet....

Cette question est très délicate à corriger. Il faut donc vérifier les points suivants.

-10 pour incohérence globale du texte, utilisation incorrecte des champs Pré/Post/Scénario etc... En particulier, -10% si les préconditions/hypothèses sont testées dans le scénario et ou si les étapes ne sont pas bien affectées à acteur ou système

Attention à la cohérence avec les dia de use case, il y a -10% si l'on ne respecte pas le dia de use case.

Erreurs fréquentes :

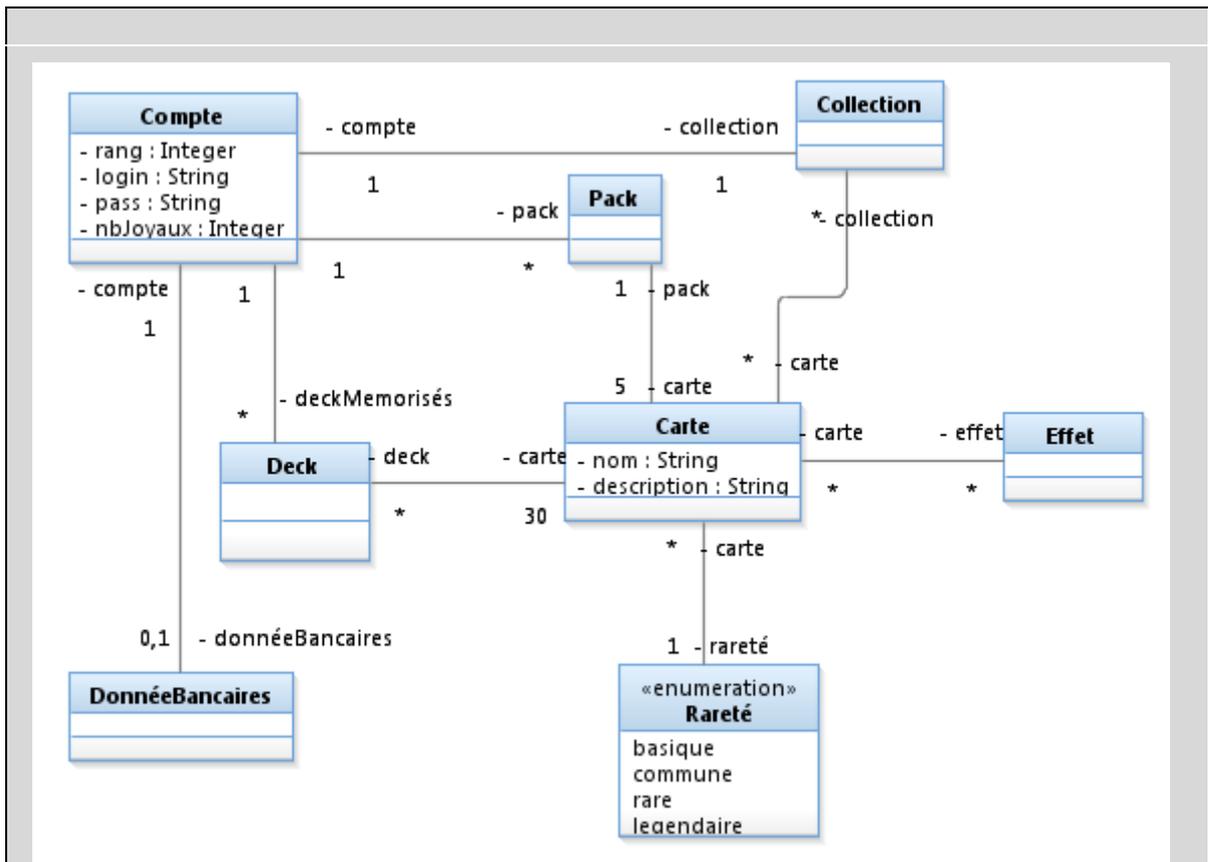
-5% à -10% on spécifie des pré/post conditions qui ne parlent de l'état du système

-10% Etapes de saisie de l'utilisateur mal distinguées des actions système, on ne sait pas clairement qui du système ou de l'acteur fait l'action dans une étape du scénario

-15% :Spécification d'étapes hors système comme étapes du scénario

Question 2.3 : (3,75 points) Réalisez le diagramme de classes métier de la phase d'analyse. Vous justifierez tous vos choix, par un texte ou des annotations sur le diagramme. Ne

modélisez pas la classe représentant le « Système », introduite dans l'approche en V du module.



Barème : (sur 110%)

Compte : avec login/pass, porte nbJoyaux et rang (15%)

Compte associé à Données bancaires : 10% si on voit un 0,1, 5% si c'est un attribut du compte

Compte associé aux decks mémorisés : 10%

Compte associé aux * Pack (pas encore ouverts) : 10%

Compte associé à * Carte : 10% (dans le corrigé via Collection mais ce n'est pas obligé)

Deck : nommé, associé à 30 cartes : 10% (on accepte * plutôt que 30)

Pack : associé à 5 cartes (* OK) 10%

Carte : avec un nom et une description, et attaque et défense (deux int) 15%

Rareté de la carte : géré via un enum 10% . On sera indulgent sur la syntaxe de l'enum + les cardinalités du lien entre cet enum et Carte. 5% si c'est juste un attribut de Carte.

Effet : lié * * à Carte : 10% (c'est du bonus).

-10% par faute grossière, méthodes, modélisation d'acteurs...

Ce n'est pas super riche, mais modéliser un peu les règles c'est trop vite compliqué.

-10% à -20% éléments dynamiques qui n'ont pas à apparaître, e.g. Acteurs.

-10% si associations orientées, compositions etc...

- 10% si opérations sur les classes
- 10% à 20% pour toute autre faute ou aberration

Question 2.4 : (2,5 pts)

A) Réalisez un diagramme de séquence de niveau analyse présentant le déroulement (scénario **nominal**) des étapes permettant à un utilisateur non connecté de vendre une de ses cartes pour obtenir des bijoux. On évitera de sur-spécifier les actions privées du système.

B) Dessinez la classe « système » afin de préciser les opérations identifiées dans cette séquence (signature, visibilité).

Essentiellement, on doit voir sur ce diagramme toute l'information circuler de l'acteur vers le système, sous une forme ou une autre. On ne doit pas nécessairement voir les opérations privées en self call

- Client se connecte (login+pass)
- Client selectionne « vendre »
- Client envoie un id de carte à vendre vers le sys
- Le sys affiche OK (boucles sur le système)

Donc :

Barème :

A) 80%

+10% lignes de vie correctes : acteur (un ou deux, on n'exige pas les deux lignes de vie du corrigé) vs système

On cherche de l'information qu'on voit circuler de l'acteur vers le système

20 % Login/pass

30% Id de la carte à détruire

20% : modélisation de vérifications ou de l'affichage en self loop sur le système, modélisation d'actions pertinentes comme « enregistrer » du système, commentaires ...

-10% on ne voit pas clairement que les lignes de vie sont des instances (notation o:Obj)

-20% si appel du système à une opération de l'acteur Agent. L'envoi asynchrone d'un message, ou une note expliquant qu'on considère que Joe représente l'acteur et son IHM => -10%. Cela reste incorrect. On cherche les responsabilités du système, pas des acteurs (donc externes au système).

-20% si on affecte des méthodes à des classes métier

B)

20% : signature(s) cohérentes avec le diag de séquence et réalisables, 0% dès qu'une incohérence est constatée. Les méthodes correspondant à d'autres use case sont tolérées mais ne donnent pas de points (hors sujet)

les self calls doivent être private (-10% si ce n'est pas le cas).

Question 2.5 : (2,5 pts) Ecrivez un test de validation couvrant l'achat d'une carte à l'aide de bijoux par un utilisateur.

* contexte : on est connecté sur "testeur", le compte dispose de fonds suffisants en bijoux (au moins 20 bijoux)

* entrée : la carte commune "Dragon vert"

* Scenario :

1. selection de l'option acheter carte
2. saisie de dragon vert dans la boite de recherche
3. selection du dragon vert
4. valider

R.A. : la carte « dragon vert » est ajoutée à la collection, la quantité de bijoux du compte est diminuée de 20.

MV: dans la construction de deck, s'assurer que la carte est disponible.

La quantité de bijoux affichée en permanence dans le coin en bas à droite est passée à 5 bijoux.

Barème :

Contexte **20%** : « on est connecté » 10%, on a les sous 10%

Entrée **10%**: champ utilisé correctement, devrait citer une carte, + éventuellement ce qu'il faut pour le scenario. : 10%. Si on met des choses qui n'habitent pas là (e.g. le prix de la carte) 0%.

Scenario **20%** : 10% cohérent avec l'objectif, étapes du testeur seulement..., 10% précision suffisante pour la reproductibilité (e.g. le testeur choisit une carte = imprécis)

R.A **20%** : carte ajoutée aux cartes du joueur 10%, bijoux débités 10%

M.V. **20%** : on vérifie la présence de la carte (10%) et le débit de bijoux (10%)

Précision numérique : **10%** le scenario cite explicitement un prix chiffré pour la carte et donc le débit, qui soit cohérent avec le CdC.

-10% à -20% aberrations énorme (alternatives dans le scenario, ...)

-10% il existe des données saisies qui ne sont pas mentionnées dans la section « entrée », ou des données dans Entrée qui ne sont pas utilisées

-20% scenario difficilement réalisable, mentionne autre que les actions utilisateur, imprécis (il doit être reproductible sans réfléchir)

-20% le scenario mentionne des actions du système (autre que résultat attendu)