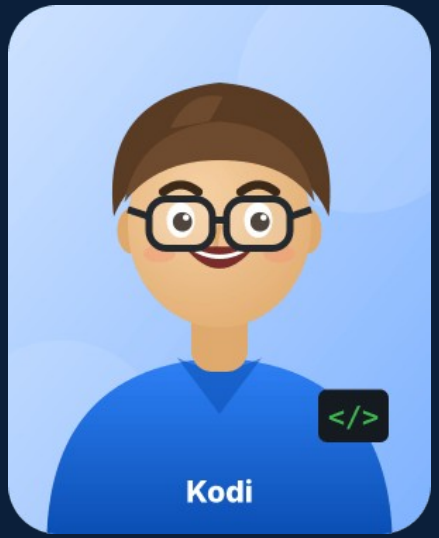
HEF-150 · MISSION 01 / 10

Product thinking

Construire un produit, ce n'est pas livrer des fonctionnalités : c'est résoudre un problème qui compte pour quelqu'un, et le prouver par des résultats.





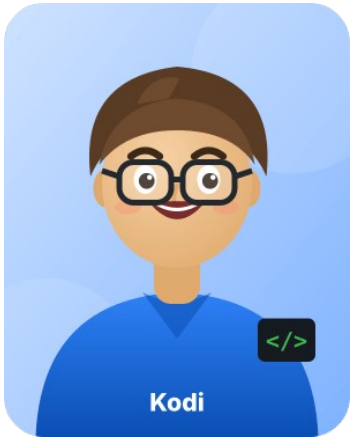
Badge : Explorateur de valeur

90 min · avancé

L'histoire

Pendant six mois, l'équipe de HashBox livre sans relâche : un module de quiz, un mode sombre, un tableau de bord pour les formateurs, des badges. Chaque démonstration est applaudie. Mais les inscriptions stagnent, et la moitié des nouveaux membres ne termine jamais son premier cours. Maya pose une question qui fait taire la réunion : « Qu'est-ce qui est meilleur *pour nos membres* qu'il y a six mois ? »

> *« Nous avons mesuré ce que nous produisons, dit Sam, pas ce que nous changeons. Une fonctionnalité est une réponse. Avant la réponse, il y a un problème, et avant le problème, une personne. Le travail de produit, c'est de relier les trois, et de vérifier qu'ils sont reliés. »*



L'histoire

L'histoire (suite)

Tu apprends à distinguer sorties et résultats, à formuler un problème avant une solution, à décrire le travail que les gens "embauchent" ton produit pour faire, et à choisir une métrique étoile et un arbre de métriques pour savoir où agir.

Ta mission

À la fin, je serai capable de…

1. distinguer produit, projet, fonctionnalité, sortie (output) et résultat (outcome)
2. formuler un problème utilisateur et une proposition de valeur indépendamment de toute solution
3. décrire un besoin avec le cadre des « tâches à accomplir » (jobs to be done)
4. construire un arbre de métriques (étoile du nord, leviers) et en faire une analyse de sensibilité
5. reconnaître les pièges : fonctionnalitis, métriques de vanité, solution cherchant un problème

Défi d'échauffement

Défi d'échauffement

Un restaurant vend des perceuses à ses clients parce que « les gens aiment les perceuses ». Personne n'en achète. Quel est le véritable problème de la personne qui entre dans une quincaillerie pour acheter une perceuse ? Cite trois façons différentes de résoudre le même besoin sans jamais vendre de perceuse. Qu'est-ce que cela change pour la façon de juger un produit ?

Le concept

Projet, produit, fonctionnalité

Notion	Définition	Question
Projet	effort limité dans le temps avec un livrable	« C'est livré à la date ? »
Produit	ce qui crée de la valeur de façon durable pour des utilisateurs et pour l'organisation	« Est-ce que ça change quelque chose pour quelqu'un ? »
Fonctionnalité	une capacité du produit	« Est-ce que ça marche ? »

Une équipe produit est responsable du **résultat** durable, pas seulement de la livraison.

Sorties et résultats

Le concept

Sorties et résultats (suite)

	Sortie (*output*)	Résultat (*outcome*)
Définition	ce que l'équipe **produit**	ce que cela **change** dans le comportement des gens
Exemple	« un tableau de bord livré »	« les formateurs corrigent 40 % plus vite »
Mesure	on livre, ou non	une métrique de comportement ou de valeur

*Un **impact** (impact) est le changement pour l'organisation (revenu, rétention). La chaîne : sorties → résultats → impact.*

Une feuille de route faite de **sorties** engendre des « usines à fonctionnalités ». Une feuille de route faite de **résultats** pose des problèmes à résoudre et laisse l'équipe chercher la meilleure solution.

Le concept

Problème avant solution

Une bonne formulation de problème est **indépendante de la solution** et **observable** :

```
[Qui] a du mal à [faire quoi] parce que [cause],
ce qui provoque [conséquence mesurable].
```

Exemple : « Les nouveaux membres ont du mal à terminer leur premier cours parce qu'ils ne savent pas par où commencer, ce qui provoque l'abandon de 50 % d'entre eux avant la deuxième semaine. »
Contre-exemple (solution déguisée) : « Il faut un tableau de bord de progression. »

La proposition de valeur

Pourquoi quelqu'un choisirait-il ton produit **plutôt qu'une alternative** (y compris « ne rien faire ») ?

Le concept

La proposition de valeur (suite)

Élément	Question
Client	qui exactement ?
Douleurs	quelles frustrations, risques, obstacles ?
Gains	quels bénéfices cherchés ?
Alternative actuelle	que fait-on aujourd'hui sans ton produit ?
Différence	pourquoi serait-ce meilleur pour lui ?

Les tâches à accomplir (JTBD)

Les gens « embauchent » un produit pour accomplir une **tâche** dans une **situation**.
Format :

Le concept

Les tâches à accomplir (JTBD) (suite)

```
Quand [situation], je veux [motivation], afin de [résultat attendu].
```

« Quand je dois mettre à jour mes compétences pour un nouveau poste, je veux un parcours qui me dit quoi apprendre dans quel ordre, afin d'être prêt en six semaines. »

La tâche a des dimensions **fonctionnelle**, **émotionnelle** (se sentir compétent) et **sociale** (être vu comme compétent). Les **concurrents** sont tout ce qui accomplit la même tâche : un tutoriel vidéo, un collègue, un cahier.

Les types de risques d'un produit

Risque	Question	Exemple
Valeur (*desirability*)	les gens en veulent-ils ?	personne n'ouvre le tableau de bord
Utilisabilité	savent-ils s'en servir ?	ils abandonnent l'inscription

Le concept

Les types de risques d'un produit (suite)

Risque	Question	Exemple
Faisabilité	savons-nous le construire ?	la correction automatique n'est pas fiable
Viabilité	est-ce rentable, légal, soutenable ?	coût par utilisateur > revenu

Le trio **produit / design / technique** examine ensemble ces quatre risques dès le début. La technique n'intervient pas « après » la décision : elle l'éclaire.

Étoile du nord et arbre de métriques

- **Étoile du nord** (*North Star*) : **une** métrique qui reflète la valeur délivrée aux utilisateurs et qui prédit le succès à long terme. Exemple : « nombre de cours terminés par semaine ».
- **Arbre de métriques** : décomposer la métrique en **leviers** (entrées) sur lesquels l'équipe peut agir : `cours terminés = membres actifs × cours démarrés par membre × taux de complétion`.

Le concept

Etoile du nord et arbre de métriques (suite)

- **Métriques de vanité** : nombre de comptes créés, de pages vues — elles montent sans prouver de valeur.

On cherche le **levier** dont l'amélioration relative a le plus d'effet et qui est le plus **améliorable** : c'est l'**analyse de sensibilité**.

Les pièges classiques

Piège	Symptôme
Usine à fonctionnalités	on mesure la livraison, pas l'effet
Solution en quête de problème	« on a une technologie, trouvons un usage »
Tout le monde est client	pas de segment précis, pas de valeur claire
Métriques de vanité	on célèbre des chiffres qui ne prouvent rien

Le concept

Les pièges classiques (suite)

Piège	Symptôme
Opinion du plus haut salaire (*HiPPO*)	la décision vient du statut, pas des preuves

🧠 ***Astuce*** *: avant toute demande de fonctionnalité, pose cinq fois « pourquoi ? » jusqu'à atteindre un comportement observable de quelqu'un. Ce comportement est le vrai point de départ.*

Exemple pas à pas

Exemple pas à pas

Problème : construire un arbre de métriques pour HashBox, calculer l'effet d'une amélioration de chaque levier sur l'étoile du nord (analyse de sensibilité), distinguer sorties et résultats dans une liste de chantiers, et vérifier qu'un problème est formulé sans solution cachée.

```
import re

# ---------------- 1. arbre de métriques ----------------
# étoile du nord : cours terminés par semaine = membres actifs × cours démarrés pa
   r membre actif × taux de complétion
ETAT = {"membres_actifs": 4000, "cours_demarres_par_membre": 0.8, "taux_completion
   ": 0.35}

def etoile_du_nord(e):
    return e["membres_actifs"] * e["cours_demarres_par_membre"] * e["taux_completi
   on"]

base = etoile_du_nord(ETAT)
print(f"étoile du nord aujourd'hui : {base:.0f} cours terminés par semaine")
```

Exemple pas à pas

Exemple pas à pas (suite)

```
assert round(base) == 1120

# ---------------- 2. sensibilité : +10 % relatif sur chaque levier --------------
   --
print("levier                          | +10 % relatif → effet sur l'étoile du nord
   ")
for levier in ETAT:
    modifie = dict(ETAT, **{levier: ETAT[levier] * 1.10})
    print(f"  {levier:28s} | {etoile_du_nord(modifie) / base - 1:+.1%}")
# le produit étant multiplicatif, +10 % sur n'importe quel levier donne +10 % : la
    sensibilité RELATIVE est la même.
# ce qui différencie les leviers, c'est la FACILITÉ d'amélioration : on compare av
   ec le gain réaliste de chacun.
GAIN_REALISTE = {"membres_actifs": 0.05, "cours_demarres_par_membre": 0.20, "taux_
   completion": 0.40}       # estimations d'équipe (hypothèses)
effets = {l: etoile_du_nord(dict(ETAT, **{l: ETAT[l] * (1 + g)})) - base for l, g
   in GAIN_REALISTE.items()}
meilleur = max(effets, key=effets.get)
print("effet réaliste (cours/semaine) :", {l: round(v) for l, v in effets.items()}
```

Exemple pas à pas

Exemple pas à pas (suite)

```
    , "→ levier prioritaire :", meilleur)
assert meilleur == "taux_completion" and round(effets["taux_completion"]) == 448

# ---------------- 3. sorties ou résultats ? ----------------
VERBES_SORTIE = ("livrer", "lancer", "créer", "ajouter", "développer", "implémente
    r", "construire", "refondre")
INDICES_RESULTAT = ("augmenter", "réduire", "%", "plus vite", "moins de", "taux",
    "passer de")

def nature(objectif):
    o = objectif.lower()
    a_resultat = any(m in o for m in INDICES_RESULTAT)
    a_sortie = o.startswith(VERBES_SORTIE)
    if a_resultat: return "résultat"
    return "sortie" if a_sortie else "à clarifier"

CHANTIERS = ["Lancer un tableau de bord de progression", "Réduire de 50 % à 30 % l
    'abandon avant la 2e semaine", "Ajouter un mode sombre",
              "Faire passer le taux de complétion du premier cours de 35 % à 45 %",
```

Exemple pas à pas (suite)

```
    "Refondre la page d'accueil", "Les formateurs corrigent 40 % plus vite"]
for c in CHANTIERS: print(f"  {nature(c):12s} ← {c}")
assert [nature(c) for c in CHANTIERS] == ["sortie", "résultat", "sortie", "résulta
   t", "sortie", "résultat"]

# ---------------- 4. un problème est-il formulé SANS solution ? ----------------
MARQUEURS_SOLUTION = ("il faut un", "il faut une", "nous devons ajouter", "il nous
    faut", "implémenter", "créer un", "créer une")
def verifier_probleme(texte):
    t = texte.lower(); remarques = []
    if any(m in t for m in MARQUEURS_SOLUTION): remarques.append("contient une sol
   ution déguisée")
    if "parce qu" not in t: remarques.append("pas de cause (« parce que »)")
    if not re.search(r"\d+\s*%|\d+", t): remarques.append("pas de conséquence mesu
   rable")
    return remarques

BON = "Les nouveaux membres ont du mal à terminer leur premier cours parce qu'ils
   ne savent pas par où commencer, ce qui provoque l'abandon de 50 % d'entre eux a
```

Exemple pas à pas

Exemple pas à pas (suite)

```
   vant la deuxième semaine."
MAUVAIS = "Il faut un tableau de bord de progression."
print("bon problème   :", verifier_probleme(BON) or "OK")
print("mauvais problème:", verifier_probleme(MAUVAIS))
assert verifier_probleme(BON) == [] and len(verifier_probleme(MAUVAIS)) == 3

# ---------------- 5. énoncé « tâche à accomplir » ----------------
def jtbd(situation, motivation, resultat):
    for nom, v in (("situation", situation), ("motivation", motivation), ("résulta
   t", resultat)):
        assert v.strip(), f"{nom} vide"
    return f"Quand {situation}, je veux {motivation}, afin de {resultat}."
print(jtbd("je dois mettre à jour mes compétences pour un nouveau poste", "un parc
   ours qui m'indique quoi apprendre et dans quel ordre", "être prêt en six semain
   es"))
```

Lecture :

Exemple pas à pas (suite)

1. L'**arbre de métriques** relie l'étoile du nord aux leviers. L'effet relatif d'une amélioration de 10 % est le même sur chaque levier d'un produit multiplicatif : ce qui compte, c'est **le gain réaliste** que l'équipe peut obtenir. Ici, améliorer le taux de complétion (de 35 % à 49 %) apporte ≈ 450 cours terminés de plus par semaine, bien plus que +5 % de membres actifs.
2. Distinguer **sorties** et **résultats** dans une liste de chantiers révèle une feuille de route faite surtout de livraisons.
3. Le vérificateur de problème détecte une **solution déguisée** (« il faut un tableau de bord »), l'absence de **cause** et de **mesure**.
4. L'énoncé **JTBD** force à écrire la situation, la motivation et le résultat attendu.

À toi, avec aide

À toi, avec aide

Un produit a pour étoile du nord « heures d'apprentissage par semaine » = `utilisateurs actifs × sessions par utilisateur × durée moyenne d'une session (heures)`.
Situation actuelle : 10 000 utilisateurs actifs, 2,5 sessions par utilisateur, 0,4 heure par session.
Trois initiatives sont envisagées, avec des gains **réalistes** estimés :

Initiative	Levier visé	Gain réaliste
Rappels par e-mail	sessions par utilisateur	+20 %
Campagne d'acquisition	utilisateurs actifs	+8 %
Contenus plus longs et plus riches	durée d'une session	+15 %

1. Calcule l'étoile du nord actuelle et l'effet de chaque initiative.
2. Classe les initiatives par effet. Lequel prioriser ?
3. Quelles autres informations te faudraient-il avant de décider (coût, risque) ?

À toi, avec aide · réponse

Indices progressifs

Indice 1

Étoile = 10 000 × 2,5 × 0,4.

Indice 2

Le gain relatif de l'étoile est égal au gain relatif du levier visé.

Indice 3

Pense au coût de chaque initiative et à la confiance dans l'estimation.

Réponse

```
base = 10000 * 2.5 * 0.4
initiatives = {"Rappels par e-mail": 0.20, "Campagne d'acquisition": 0.08, "Conten
   us plus longs et plus riches": 0.15}
for nom, gain in sorted(initiatives.items(), key=lambda kv: -kv[1]):
    print(f"{nom:36s} +{base * gain:6.0f} heures/semaine ({gain:+.0%})")
assert base == 10000 and round(base * 0.20) == 2000
```

1. Étoile du nord actuelle : **10 000 heures par semaine**.
2. Classement : rappels par e-mail (+2 000 h), contenus plus riches (+1 500 h), campagne d'acquisition (+800 h). Les rappels semblent prioritaires.
3. Il manque : le **coût** de chaque initiative (une campagne coûte de l'argent, des contenus riches des mois de production), la **confiance** dans chaque estimation (le gain de +20 % est-il fondé sur des données ou un vœu ?), les **risques** (des rappels peuvent agacer et réduire la rétention), et l'**effet sur l'impact** (heures d'apprentissage ≠ valeur pour l'utilisateur si elles ne conduisent pas à terminer). C'est l'objet des missions de priorisation (7) et d'hypothèses (3).

À toi, en solo

Clarificateur de feuille de route. Écris un outil qui transforme une feuille de route de sorties en feuille de route de résultats :

1. une structure `Chantier` avec : `titre`, `type` (sortie/résultat), `probleme` (texte), `metrique_cible` (nom), `valeur_actuelle`, `valeur_visee`, `echeance` ;
2. une fonction `audit(chantiers)` qui, pour chaque chantier, renvoie des remarques : sortie sans problème associé, résultat sans valeur actuelle ou visée chiffrée, objectif visé inférieur ou égal à l'actuel pour une métrique « plus est mieux », problème contenant une solution déguisée ;
3. un score de « maturité produit » de la feuille de route : part des chantiers qui sont des résultats chiffrés et reliés à une métrique de l'arbre ;
4. teste avec une feuille de route de 8 chantiers : 5 sorties pures, 3 résultats dont un mal chiffré ; vérifie le score avant, puis réécris les chantiers et vérifie qu'il monte.

À toi, en solo · réponse

Pistes de correction

```
from dataclasses import dataclass
from typing import Optional

METRIQUES_ARBRE = {"taux_completion", "membres_actifs", "cours_demarres_par_membre
   ", "retention_s2"}
SOLUTIONS = ("il faut un", "il faut une", "ajouter un", "ajouter une", "créer un",
    "créer une")

@dataclass
class Chantier:
    titre: str
    type: str                                  # "sortie" ou "résultat"
    probleme: str = ""
    metrique_cible: str = ""
    valeur_actuelle: Optional[float] = None
    valeur_visee: Optional[float] = None
    echeance: str = ""

def audit(chantiers):
```

À toi, en solo · réponse

Pistes de correction (suite)

```
    remarques = {}
    for c in chantiers:
        r = []
        if c.type == "sortie" and not c.probleme: r.append("sortie sans problème a
   ssocié")
        if c.probleme and any(m in c.probleme.lower() for m in SOLUTIONS): r.appen
   d("problème = solution déguisée")
        if c.type == "résultat":
            if c.metrique_cible not in METRIQUES_ARBRE: r.append("métrique hors de
    l'arbre de métriques")
            if c.valeur_actuelle is None or c.valeur_visee is None: r.append("vale
   urs actuelle / visée manquantes")
            elif c.valeur_visee <= c.valeur_actuelle: r.append("objectif inférieur
    ou égal à l'existant")
            if not c.echeance: r.append("pas d'échéance")
        remarques[c.titre] = r
    return remarques

def maturite(chantiers):
```

À toi, en solo · réponse

Pistes de correction (suite)

```
    ok = sum(1 for c in chantiers if c.type == "résultat" and not audit([c])[c.tit
   re])
    return ok / len(chantiers)

avant = [Chantier("Tableau de bord de progression", "sortie"), Chantier("Mode somb
   re", "sortie"), Chantier("Refonte de l'accueil", "sortie"),
         Chantier("Badges", "sortie", probleme="Il faut un système de badges"), Ch
   antier("Application mobile", "sortie"),
         Chantier("Améliorer la complétion", "résultat", metrique_cible="taux_comp
   letion", valeur_actuelle=0.35, valeur_visee=0.45, echeance="T3"),
         Chantier("Réduire l'abandon", "résultat", metrique_cible="retention_s2",
   valeur_actuelle=0.5),
         Chantier("Plus de visites", "résultat", metrique_cible="pages_vues", vale
   ur_actuelle=100, valeur_visee=150, echeance="T3")]
m_avant = maturite(avant)
print(f"maturité avant : {m_avant:.0%}")
for t, r in audit(avant).items():
    if r: print("  ", t, "→", r)
apres = [Chantier("Aider les nouveaux à démarrer", "résultat", "ils ne savent pas
```

À toi, en solo · réponse

Pistes de correction (suite)

```
   par où commencer", "taux_completion", 0.35, 0.45, "T3"),
         Chantier("Réduire l'abandon de la semaine 2", "résultat", "ils perdent le
    fil sans rappel", "retention_s2", 0.5, 0.65, "T3"),
         Chantier("Augmenter les cours démarrés", "résultat", "le catalogue est di
   fficile à parcourir", "cours_demarres_par_membre", 0.8, 1.0, "T4"),
         Chantier("Mode sombre", "sortie", "lecture de nuit inconfortable")]
m_apres = maturite(apres)
print(f"maturité après : {m_apres:.0%}")
assert m_avant == 1 / 8 and m_apres == 3 / 4
```

L'outil rend visible le **ratio sorties / résultats** de la feuille de route et rappelle que chaque résultat doit être **chiffré**, **daté**, et rattaché à une métrique de l'arbre. Une sortie reste légitime (un mode sombre), mais elle doit être reliée à un problème.

Pièges classiques

Piège	Pourquoi c'est un problème	Comment l'éviter
Mesurer la livraison, pas l'effet	Une usine à fonctionnalités qui ne change rien	Définir des résultats chiffrés par comportement
Solution déguisée en problème	On construit avant de comprendre	Formuler Qui / difficulté / cause / conséquence mesurable
Métriques de vanité	Des chiffres qui montent sans prouver de valeur	Étoile du nord + arbre de leviers actionnables
« Tout le monde » est le client	Aucune proposition de valeur précise	Segment, tâche à accomplir, alternative actuelle

Mini-boss

Mini-boss

Cadrage produit d'une idée. Choisis une idée de produit (la tienne ou « application de révision pour les cours HashCode ») et écris son cadrage complet sous forme de données et de code :

1. **problème** formulé sans solution, **segment** précis, **tâche à accomplir** (JTBD), **alternatives actuelles** ;
2. **proposition de valeur** (douleurs, gains, différence) ;
3. **quatre risques** (valeur, utilisabilité, faisabilité, viabilité) avec pour chacun la question à tester ;
4. **étoile du nord** et **arbre de métriques** (3 à 4 leviers) avec valeurs de départ fictives et gains réalistes ; calcule le levier prioritaire ;
5. une **feuille de route de résultats** (4 chantiers chiffrés) évaluée par ton `audit` (aucune remarque) ;
6. un vérificateur automatique de cohérence (problème sans solution, chaque risque a une question, chaque chantier relié à un levier) ;

Mini-boss

Mini-boss (suite)

7. 10 lignes : ce que tu **ne sais pas encore** et qui pourrait invalider le cadrage.

Mini-boss · réponse

Pistes de correction

Un bon cadrage : problème et segment concrets, JTBD à trois champs, tableau des quatre risques avec questions testables (« 30 % des personnes interrogées ont-elles réellement acheté une solution payante au cours des 6 derniers mois ? »), arbre de métriques multiplicatif et analyse du levier prioritaire, feuille de route de résultats datés, vérificateur qui échoue si l'on glisse une solution dans le problème. La liste des inconnues est honnête (je ne sais pas si le problème est assez douloureux pour payer).

Quiz éclair

Quiz éclair

1. Un résultat (outcome) est :

A. un changement observable dans le comportement des utilisateurs
B. une fonctionnalité livrée
C. un fichier de spécifications

2. Un énoncé de problème bien formé :

A. décrit l'écran à construire
B. ne contient pas de solution et mentionne qui, quoi, pourquoi, et une conséquence mesurable
C. est une liste de tâches

3. L'étoile du nord est :

A. le nombre de lignes de code

Quiz éclair

Quiz éclair (suite)

B. le chiffre d'affaires du trimestre précédent

C. une métrique qui reflète la valeur délivrée aux utilisateurs

4. 🔁 *Révision HEF-140, mission 10* : Un invariant de sûreté est :

A. une propriété qui doit être vraie dans toute exécution, par exemple « jamais plus de N inscrits »

B. un objectif de vitesse

C. une fonctionnalité optionnelle

5. 🔁 *Révision HEF-140, mission 09* : Un jeton de barrière (fencing token) empêche :

A. la perte de paquets

B. l'écriture d'un ancien détenteur d'un bail expiré

Quiz éclair

Quiz éclair (suite)

C. les doublons de messages

Quiz éclair · réponse

Réponses

1. **A**.
2. **B**.
3. **C**.
4. **A**.
5. **B**.

Je réfléchis

1. Quel chantier de ton équipe est une sortie dont tu ne connais pas le résultat attendu ?
2. Peux-tu formuler en une phrase, sans solution, le problème de ton utilisateur principal ?

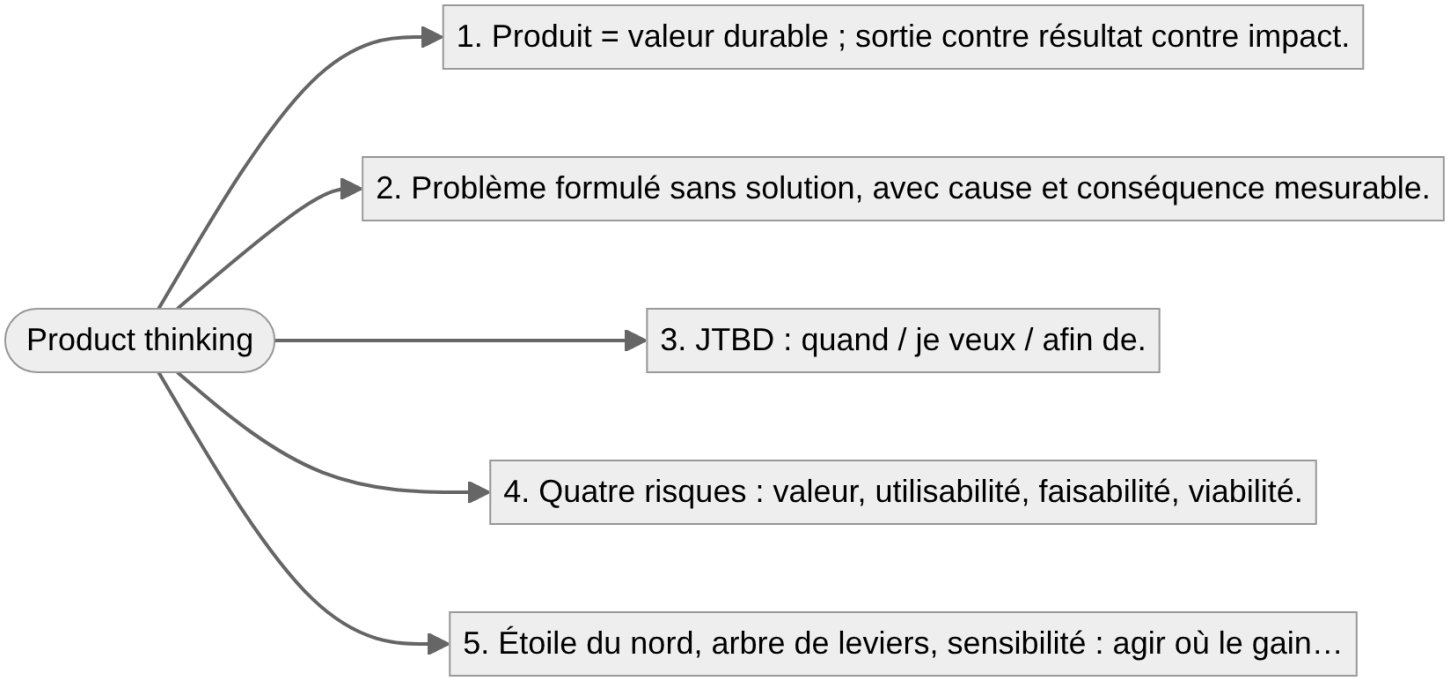
À retenir

À retenir

1. Produit = valeur durable ; sortie contre résultat contre impact.
2. Problème formulé sans solution, avec cause et conséquence mesurable.
3. JTBD : quand / je veux / afin de.
4. Quatre risques : valeur, utilisabilité, faisabilité, viabilité.
5. Étoile du nord, arbre de leviers, sensibilité : agir où le gain réaliste est le plus fort.

À retenir

Le schéma de la mission



MISSION ACCOMPLIE

+100 XP

Badge débloqué : Explorateur de valeur

Prochaine mission : Discovery et problèmes utilisateurs

