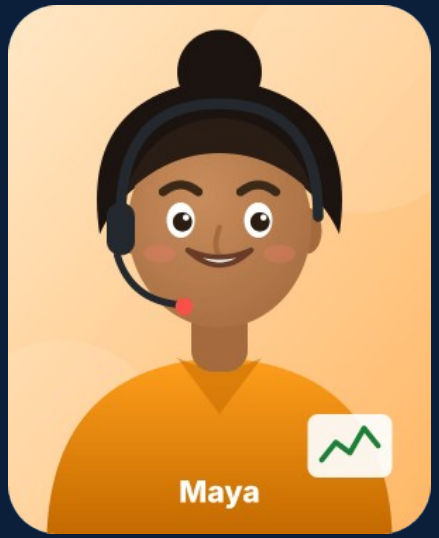
HEF-150 · MISSION 10 / 10

Revue de décision produit



Après toutes les hypothèses, mesures et expériences, il faut décider : persévérer, pivoter ou arrêter. La qualité d'une décision se juge à la façon dont elle est prise, pas seulement à son résultat.

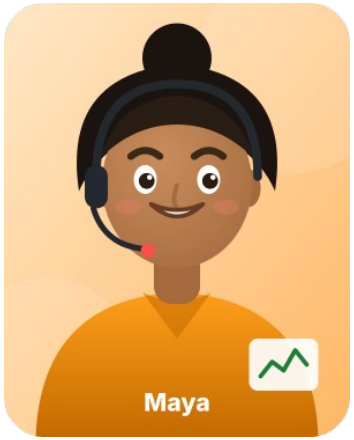


Badge : Décideur éclairé

130 min · avancé

L'histoire

Quatre mois après le lancement du MVP de l'espace formateurs, la direction réunit l'équipe : « Nous avons investi 180 000 €. Les chiffres sont mitigés : 9 formateurs actifs au lieu des 25 visés, mais un d'entre eux a généré 40 % du revenu. Que fait-on ? » Le chef de projet plaide : « On a trop investi pour arrêter. » Le commercial : « Donnons trois mois de plus. » La fondatrice : « J'ai toujours cru à ce produit. » Personne ne regarde les critères que l'équipe avait fixés au départ.



« Trois biais dans une seule phrase, constate Maya : le coût irrécupérable, l'espoir sans date de fin, et la conviction personnelle. Une revue de décision sert à faire autrement : relire les critères fixés avant, regarder les preuves avec leurs incertitudes, calculer ce que vaut l'avenir de chaque option — jamais le passé — et écrire la décision et ses raisons pour apprendre, même si elle est mauvaise. »

L'histoire

L'histoire (suite)

Dans cette dernière mission, tu conduis une revue de décision produit complète : critères préétablis, évidences, arbre de décision, valeur de l'information, biais, mémo de décision et calibration des prévisions.

Ta mission

À la fin, je serai capable de…

1. évaluer des hypothèses par rapport à des critères fixés à l'avance, en tenant compte de l'incertitude
2. comparer persévérer, pivoter et arrêter par la valeur espérée de l'avenir (sans coût irrécupérable)
3. calculer la valeur d'une information supplémentaire avant de décider
4. repérer et neutraliser les biais de décision, avec une pré-mortem
5. rédiger un mémo de décision et suivre la calibration de ses propres prévisions

Défi d'échauffement

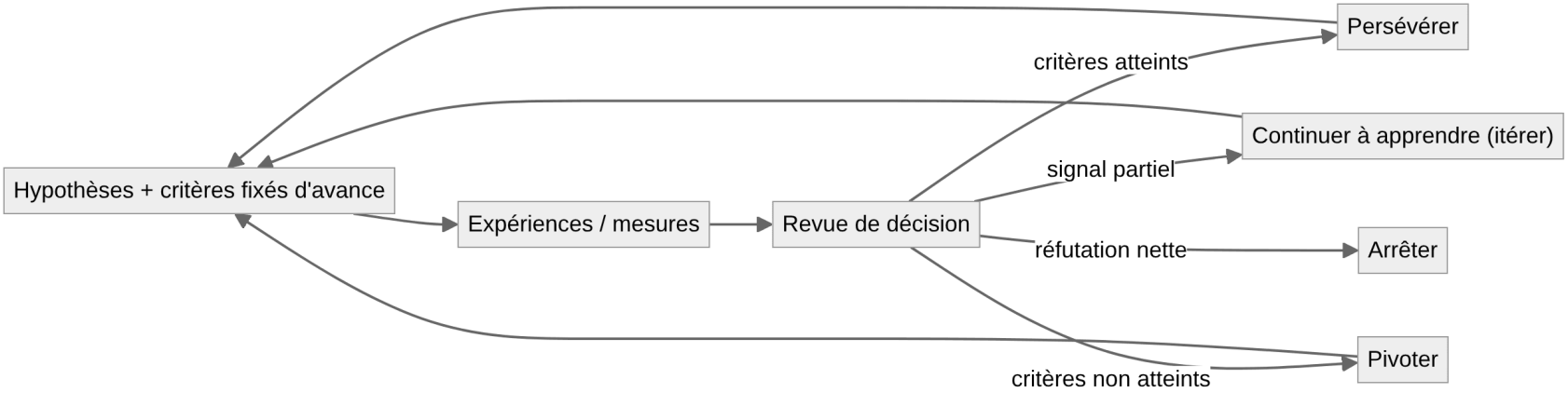
Défi d'échauffement

Tu as parié sur un cheval avec 100 €. À mi-course, le cheval est dernier et le jockey est blessé. On te propose d'ajouter 50 € « pour te refaire » car « tu as déjà tant misé ». Qu'est-ce qui compte dans la décision d'ajouter ou non les 50 € : les 100 € déjà perdus, ou les chances actuelles ? Écris en une phrase la bonne façon de raisonner, puis l'erreur qui consiste à raisonner autrement.

Le concept

Le cycle de décision produit

Les mêmes questions reviennent à chaque cycle (mission 3) :



Le concept

Le cycle de décision produit (suite)

Description du schéma : à partir d'hypothèses et de critères fixés d'avance, des expériences produisent des mesures ; la revue de décision conduit à persévérer, itérer, pivoter ou arrêter ; sauf arrêt, on repart sur de nouvelles hypothèses.

Les trois décisions

Décision	Sens	Signal typique
Persévérer	continuer dans la même direction, souvent en élargissant	critères de succès atteints avec marge ; intervalle de confiance au-dessus du seuil
Pivoter	changer une hypothèse fondamentale (segment, problème, solution, canal, modèle économique) en gardant l'apprentissage	résultats insuffisants mais **signaux** qu'une variante pourrait marcher
Arrêter	cesser d'investir	intervalle de confiance **sous** le seuil d'échec, ou meilleur usage du temps ailleurs

Le concept

Les trois décisions (suite)

Arrêter à temps est une **réussite** : on libère la capacité pour une meilleure idée.

Les critères préétablis

Dès l'étape d'hypothèse (mission 3), on a fixé : **métrique**, **seuil de succès**, **seuil d'échec**, **horizon**. La revue **relit ces critères** avant de regarder les résultats, puis applique la règle. Déplacer les poteaux après le match est la forme de fraude intellectuelle la plus courante.

Évaluer l'évidence

Pour chaque hypothèse : le résultat **mesuré**, son **intervalle** (mission 3), la **taille d'échantillon**, les **biais** possibles (recrutement, effet de nouveauté, saisonnalité). Quatre niveaux de preuve :

Niveau	Source
1	opinions, anecdotes

Le concept

Évaluer l'évidence (suite)

Niveau	Source
2	entretiens, observations
3	comportements mesurés (usage, achats)
4	expérience contrôlée (A/B)

Une décision à fort enjeu exige du niveau 3 ou 4, pas du niveau 1.

Les coûts irrécupérables

Ce qui est **déjà dépensé** (180 000 €) ne doit **pas** entrer dans la décision : on ne peut pas le récupérer, quelle que soit la suite. La bonne question est : *« à partir de maintenant, quelle option a la meilleure valeur espérée ? »* La **valeur** inclut ce qu'on ne fera pas pendant ce temps (**coût d'opportunité**).

Le concept

Arbre de décision et valeur espérée

Pour chaque option : liste des **issues**, leur **probabilité** (avec fourchette), leur **gain net** (revenu − coût futur). `VE = Σ p × gain`. On compare les VE, puis on regarde la **variance** (le pire cas), la **réversibilité** et l'**option de revenir plus tard**.

Option	Issues
Persévérer (6 mois, 60 k€)	succès 25 % : +400 k€ ; échec 75 % : 0
Pivoter (4 mois, 40 k€)	succès 35 % : +300 k€ ; échec 65 % : 0
Arrêter	0 (capacité réaffectée : +40 k€ de valeur attendue ailleurs)

Le concept

Valeur de l'information

Parfois, la meilleure décision est d'**attendre un peu et apprendre**. La **valeur de l'information parfaite** : `VEIP = E[meilleure décision connaissant l'issue] − max VE`. C'est le **plafond** de ce qu'il vaut la peine de payer pour une expérience : si une expérience coûte plus que ce plafond, ne la fais pas.

Les biais de décision

Biais	Phrase typique	Remède
Coût irrécupérable	« on a trop investi pour arrêter »	ne regarder que l'avenir
Escalade d'engagement	« encore trois mois » sans critère	horizon et seuil d'arrêt écrits

Le concept

Les biais de décision (suite)

Biais	Phrase typique	Remède
Confirmation	on ne retient que ce qui plaît	chercher des preuves contraires, avocat du diable
Optimisme / planification	« ce sera fini en deux mois »	prévision de référence (historique), Monte Carlo
Ancrage	le premier chiffre domine	estimations indépendantes
HiPPO	l'avis du plus haut placé	avis écrits avant la discussion
Résultat plutôt que processus	« ça a marché, donc c'était bien »	évaluer la qualité de la décision à son information du moment

Pré-mortem

Le concept

Pré-mortem (suite)

Avant de s'engager : *« imaginons que dans un an ce projet a échoué. Quelles en sont les raisons ? »* Chacun écrit ses causes en silence ; on regroupe ; on traite les plus probables (surveillance, mitigation, seuil d'arrêt). Il libère la parole sur les risques.

La qualité d'une décision et la calibration

On ne peut pas juger une décision à son seul résultat (chance, variance). On juge : l'**information utilisée**, les **options** considérées, les **valeurs** pesées, le **raisonnement**. Pour s'améliorer, on tient un **journal de décisions** avec des **prévisions chiffrées** (« 60 % de chances d'atteindre 20 formateurs ») et on mesure la **calibration** (score de Brier : `(p − résultat)²`, plus bas = mieux ; sur 100 prévisions à 70 %, ~70 doivent se réaliser).

Le mémo de décision

Court, écrit, daté :

Le concept

Le mémo de décision (suite)

1. **Contexte** et décision à prendre ;
2. **Critères préétablis** et **résultats** avec incertitude ;
3. **Options** et **valeur espérée** ;
4. **Décision**, raisons, **ce qu'on arrête** ;
5. **Risques**, pré-mortem, **signaux** qui nous feraient reconsidérer ;
6. **Prévisions chiffrées** pour la prochaine revue ;
7. Responsable et date de la prochaine revue.

🧠 ***Astuce*** *: écris la décision avant la réunion, avec tes raisons. Après la discussion, compare : qu'est-ce qui a changé ton avis, et était-ce de l'information ou de la pression ?*

Exemple pas à pas

Exemple pas à pas

Problème : appliquer des critères préétablis aux résultats d'un MVP (avec intervalles), comparer persévérer / pivoter / arrêter par valeur espérée sans coût irrécupérable, calculer la valeur de l'information, mesurer la calibration par le score de Brier, et générer le mémo de décision.

```
import math, random

# ---------------- 1. critères préétablis contre résultats mesurés ---------------
    -
CRITERES = [   # (nom, seuil de succès, seuil d'échec, plus est mieux ?)
    ("formateurs actifs à 12 semaines", 20, 8, True),
    ("part des formateurs publiant un 2e cours", 0.40, 0.15, True),
    ("coût d'acquisition d'un formateur (€)", 300, 800, False),
]
RESULTATS = {   # valeur mesurée et son intervalle de confiance (bas, haut)
    "formateurs actifs à 12 semaines": (9, (7, 12)),
    "part des formateurs publiant un 2e cours": (0.44, (0.19, 0.72)),
    "coût d'acquisition d'un formateur (€)": (410, (320, 560)),
}
```

Exemple pas à pas

Exemple pas à pas (suite)

```
def verdict_critere(nom, succes, echec, plus_mieux, valeur, ic):
    bas, haut = ic
    if not plus_mieux: succes, echec, bas, haut = -succes, -echec, -haut, -bas
            # on se ramène à « plus est mieux »
    if bas >= succes: return "atteint (même le bas de l'intervalle)"
    if haut <= echec: return "échec net (même le haut de l'intervalle)"
    if haut < succes: return "non atteint, mais l'échec n'est pas démontré"
    return "incertain : l'intervalle contient le seuil de succès"
for nom, s, e, pm in CRITERES:
    valeur, ic = RESULTATS[nom]; print(f"  {nom:42s}: {valeur} {ic} → {verdict_cri
   tere(nom, s, e, pm, valeur, ic)}")
assert verdict_critere(*CRITERES[0], *RESULTATS[CRITERES[0][0]]).startswith("non a
   tteint")

# ---------------- 2. persévérer, pivoter, arrêter : valeur espérée de L'AVENIR --
   --------------
COUT_IRRECUPERABLE = 180000                                          # déjà
   dépensé : N'ENTRE PAS dans le calcul
OPTIONS = {
```

Exemple pas à pas (suite)

```
    "persévérer (6 mois, 60 k€)": dict(cout=60000, issues=[(0.25, 400000), (0.75,
   0)]),
    "pivoter (4 mois, 40 k€)":    dict(cout=40000, issues=[(0.35, 300000), (0.65,
   0)]),
    "arrêter (capacité réaffectée)": dict(cout=0, issues=[(1.0, 40000)]),
}
def valeur_esperee(o): return sum(p * g for p, g in o["issues"]) - o["cout"]
for nom, o in OPTIONS.items(): print(f"  VE de « {nom} » = {valeur_esperee(o):+9.0
   f} €")
meilleure = max(OPTIONS, key=lambda n: valeur_esperee(OPTIONS[n])); print("meilleu
   re option :", meilleure)
assert meilleure.startswith("pivoter")
# la même décision AVEC le coût irrécupérable ajouté à chaque option ne change rie
   n : il se soustrait de TOUTES les options (c'est un biais d'y voir un argument)
assert max(OPTIONS, key=lambda n: valeur_esperee(OPTIONS[n]) - COUT_IRRECUPERABLE)
    == meilleure

# ---------------- 3. sensibilité aux probabilités ----------------
def meilleure_option(p_persev, p_pivot):
```

Exemple pas à pas

Exemple pas à pas (suite)

```
    opts = {"persévérer": p_persev * 400000 - 60000, "pivoter": p_pivot * 300000 -
    40000, "arrêter": 80000}
    return max(opts, key=opts.get)
rng = random.Random(1); victoires = {"persévérer": 0, "pivoter": 0, "arrêter": 0}
for _ in range(10000):
    victoires[meilleure_option(rng.uniform(0.10, 0.40), rng.uniform(0.20, 0.50))]
   += 1                  # probabilités incertaines
print("part des cas où chaque option est la meilleure :", {k: f"{v / 100:.0f} %" f
   or k, v in victoires.items()})
assert sum(victoires.values()) == 10000

# ---------------- 4. valeur de l'information parfaite ----------------
# deux états du monde : « le segment des formateurs indépendants est réceptif » (p
    = 0,35) ou non
P_RECEPTIF = 0.35
GAINS = {   # gain net de chaque option selon l'état du monde
    "persévérer": {"réceptif": 600000 - 60000, "non": 0 - 60000},
    "pivoter": {"réceptif": 350000 - 40000, "non": 50000 - 40000},
    "arrêter": {"réceptif": 80000, "non": 80000},
```

Exemple pas à pas

Exemple pas à pas (suite)

```
}
def ve(option): return P_RECEPTIF * GAINS[option]["réceptif"] + (1 - P_RECEPTIF) *
    GAINS[option]["non"]
meilleure_sans_info = max(GAINS, key=ve)
ve_avec_info_parfaite = P_RECEPTIF * max(g["réceptif"] for g in GAINS.values()) +
   (1 - P_RECEPTIF) * max(g["non"] for g in GAINS.values())
veip = ve_avec_info_parfaite - ve(meilleure_sans_info)
print(f"meilleure option sans information : {meilleure_sans_info} (VE {ve(meilleur
   e_sans_info):,.0f} €) ; avec information parfaite : {ve_avec_info_parfaite:,.0f
   } € → valeur de l'information ≤ {veip:,.0f} €")
assert veip > 0
for cout_experience in (20000, 150000): print(f"  une expérience à {cout_experienc
   e:,} € {'vaut la peine' if cout_experience < veip else 'ne vaut PAS la peine'}
   (plafond {veip:,.0f} €)")

# ---------------- 5. calibration : score de Brier ----------------
def brier(previsions): return sum((p - r) ** 2 for p, r in previsions) / len(previ
   sions)
def table_calibration(previsions, tranches=(0.0, 0.3, 0.5, 0.7, 0.9, 1.01)):
```

Exemple pas à pas (suite)

```
    lignes = []
    for a, b in zip(tranches, tranches[1:]):
        s = [(p, r) for p, r in previsions if a <= p < b]
        if s: lignes.append((f"{a:.0%}-{min(b, 1):.0%}", len(s), sum(p for p, _ in
    s) / len(s), sum(r for _, r in s) / len(s)))
    return lignes
rng = random.Random(2)
def previsions_equipe(biais, n=300):                                    # bi
   ais > 0 : sur-confiance
    res = []
    for _ in range(n):
        vrai = rng.uniform(0.1, 0.9); r = 1 if rng.random() < vrai else 0
        dit = min(0.99, max(0.01, 0.5 + (vrai - 0.5) * (1 + biais)))
        res.append((dit, r))
    return res
for nom, biais in (("calibrée", 0.0), ("sur-confiante", 1.2)):
    pv = previsions_equipe(biais); print(f"équipe {nom:14s} : Brier = {brier(pv):.
   3f}")
    if nom == "sur-confiante":
```

Exemple pas à pas (suite)

```
        for tranche, n, dit, vu in table_calibration(pv): print(f"     {tranche:9s
   } n = {n:3d} | annoncé {dit:.0%} | réalisé {vu:.0%}")
assert brier(previsions_equipe(0.0)) < brier(previsions_equipe(1.2))

# ---------------- 6. mémo de décision ----------------
def memo(titre, decision, raisons, criteres, options, risques, previsions, respons
   able, prochaine_revue):
    l = [f"# Mémo de décision — {titre}", "", f"**Décision : {decision}**", "", "#
   # Critères préétablis et résultats"]
    l += [f"- {nom} : {RESULTATS[nom][0]} {RESULTATS[nom][1]} → {verdict_critere(n
   om, s, e, pm, *RESULTATS[nom])}" for nom, s, e, pm in criteres]
    l += ["", "## Options (valeur espérée de l'avenir, coûts irrécupérables exclus
   )"]
    l += [f"- {n} : {valeur_esperee(o):+,.0f} €" for n, o in options.items()]
    l += ["", "## Raisons"] + [f"- {r}" for r in raisons] + ["", "## Risques et si
   gnaux de réexamen"] + [f"- {r}" for r in risques]
    l += ["", "## Prévisions chiffrées pour la prochaine revue"] + [f"- {p}" for p
    in previsions] + ["", f"Responsable : {responsable} — prochaine revue : {proch
   aine_revue}"]
```

Exemple pas à pas (suite)

```
    return "\n".join(l)
texte = memo("Espace formateurs", "PIVOTER vers les organismes de formation (segme
   nt B2B)", ["la valeur espérée de l'avenir est la plus haute", "un formateur gén
   ère 40 % du revenu : le segment organisme est plus rentable", "le coût irrécupé
   rable (180 k€) n'a pas été pris en compte"],
            CRITERES, OPTIONS, ["échec du pivot si moins de 3 organismes signent
   à 8 semaines → arrêter", "dépendance à un client majeur"], ["P(≥ 3 organismes s
   ignés en 8 semaines) = 40 %", "P(coût d'acquisition < 600 €) = 55 %"], "Maya",
   "12 semaines")
print(texte)
assert "coûts irrécupérables exclus" in texte and "prochaine revue" in texte and "
   Prévisions chiffrées" in texte
```

Lecture :

Exemple pas à pas

Exemple pas à pas (suite)

1. Les **critères préétablis** sont appliqués avec les **intervalles** : le nombre de formateurs actifs (9) est sous le seuil de succès, mais l'intervalle exclut-il l'échec net ? Ici, on ne conclut pas « tout va bien » ni « c'est fini » : chaque critère reçoit son verdict honnête, y compris « incertain ».
2. Le **coût irrécupérable** (180 000 €) est affiché mais **n'entre pas** dans la comparaison : le soustraire de toutes les options ne change pas le classement. L'argument « on a trop investi pour arrêter » est donc un biais.
3. La **sensibilité** aux probabilités montre quand le classement bascule : quand les probabilités sont incertaines, l'option gagnante n'est pas toujours la même — d'où l'intérêt de **réduire l'incertitude** si c'est peu coûteux.
4. La **valeur de l'information parfaite** plafonne ce qu'il vaut la peine de payer pour une expérience : une étude à 20 000 € est justifiée, une à 150 000 € non.
5. Le **score de Brier** et la table de calibration révèlent qu'une équipe sur-confiante annonce 90 % là où elle ne réalise que 65 %.

Exemple pas à pas (suite)

6. Le **mémo de décision** consigne critères, options, raisons, risques et prévisions : la prochaine revue pourra comparer.

À toi, avec aide

À toi, avec aide

Une équipe doit décider du sort d'un produit. Estimations :

Option	Coût futur	Issues (probabilité → gain brut)
Continuer	50 000 €	20 % → 350 000 € ; 80 % → 0
Pivoter	30 000 €	30 % → 250 000 € ; 70 % → 20 000 €
Arrêter	0	100 % → 60 000 € (équipe réaffectée)

Le produit a déjà coûté 200 000 €.

1. Calcule la valeur espérée de chaque option (gain brut espéré − coût futur).
2. Quelle option choisis-tu ? Le coût déjà dépensé change-t-il le classement ?
3. Un responsable dit : « Si on arrête, on perd 200 000 € ». Que lui réponds-tu ?

À toi, avec aide (suite)

4. À partir de quelle probabilité de succès pour « Continuer » cette option devient-elle la meilleure ?

À toi, avec aide · réponse

Indices progressifs

Indice 1

VE = Σ p × gain − coût.

Indice 2

Le coût déjà dépensé s'applique à toutes les options.

Indice 3

Cherche p tel que 350 000·p − 50 000 = meilleure VE des autres options.

À toi, avec aide · réponse

Réponse

```
OPT = {"Continuer": (50000, [(0.2, 350000), (0.8, 0)]), "Pivoter": (30000, [(0.3,
   250000), (0.7, 20000)]), "Arrêter": (0, [(1.0, 60000)])}
ve = {n: sum(p * g for p, g in iss) - c for n, (c, iss) in OPT.items()}
print({n: round(v) for n, v in ve.items()})
meilleur_autre = max(ve["Pivoter"], ve["Arrêter"])
p_seuil = (meilleur_autre + 50000) / 350000
print(f"« Continuer » devient la meilleure option si P(succès) > {p_seuil:.1%}")
assert max(ve, key=ve.get) == "Arrêter" and round(ve["Pivoter"]) == 59000 and ve["
   Continuer"] == 20000 and round(p_seuil, 3) == 0.314
```

1. **Continuer : +20 000 €**, **Pivoter : +59 000 €**, **Arrêter : +60 000 €**.
2. **Arrêter** (60 000 €) est à peine meilleur que **Pivoter** (59 000 €) : les deux sont proches, donc la décision dépend des **risques** et de la **réversibilité**. Le coût déjà dépensé (200 000 €) est le même quelle que soit l'option : il ne change **pas** le classement.

Réponse (suite)

3. « Ce qui est dépensé est perdu dans tous les cas. La seule question est ce que nous gagnons **à partir de maintenant**. Continuer à investir pour récupérer le passé, c'est dépenser 50 000 € pour une valeur espérée inférieure à ce que nous obtiendrions en arrêtant. »
4. « Continuer » devient la meilleure option si la probabilité de succès dépasse **31 %** (au lieu des 20 % estimés) : si une expérience peu coûteuse pouvait relever cette probabilité, elle aurait de la valeur.

À toi, en solo

Journal de décisions et calibration. Écris un petit outil de suivi des décisions :

1. une structure `Decision` : titre, date, options, option choisie, raisons, `previsions` (liste de (énoncé, probabilité)), échéance de vérification ;
2. un enregistreur de résultats : pour chaque prévision, issue vraie/fausse à l'échéance ;
3. un calcul de **score de Brier** global et par personne, et une **table de calibration** par tranche ;
4. un détecteur de biais : sur-confiance (probabilités moyennes annoncées supérieures aux réalisations de plus de 10 points), sous-confiance, ancrage (prévisions toutes identiques) ;
5. un rapport de qualité de décision séparant **processus** (critères préétablis, options ≥ 3, pré-mortem fait, coûts irrécupérables exclus) et **résultat** ;
6. teste avec 60 prévisions simulées pour 3 personnes (une sur-confiante, une calibrée, une qui répond toujours 50 %) et vérifie que l'outil les distingue.

À toi, en solo · réponse

Pistes de correction

```
import random
from collections import defaultdict
from dataclasses import dataclass, field

@dataclass
class Decision:
    titre: str
    options: list
    choisie: str
    raisons: list
    criteres_preetablis: bool = False
    pre_mortem: bool = False
    couts_irrecuperables_exclus: bool = False
    previsions: list = field(default_factory=list)          # (personne, énoncé, p
   robabilité, résultat ou None)

    def processus(self):
        points = {"critères préétablis": self.criteres_preetablis, "au moins 3 opt
   ions": len(self.options) >= 3,
```

Pistes de correction (suite)

```
                "pré-mortem fait": self.pre_mortem, "coûts irrécupérables exclus
   ": self.couts_irrecuperables_exclus, "raisons écrites": len(self.raisons) >= 2}
        return sum(points.values()) / len(points), [k for k, v in points.items() i
   f not v]

def brier(pv): return sum((p - r) ** 2 for p, r in pv) / len(pv)
def calibration(pv):
    t = defaultdict(list)
    for p, r in pv: t[min(int(p * 5), 4)].append((p, r))
    return {f"{i * 20}-{i * 20 + 20}%": (len(v), sum(p for p, _ in v) / len(v), su
   m(r for _, r in v) / len(v)) for i, v in sorted(t.items())}
def biais(pv):
    moyenne_annoncee = sum(p for p, _ in pv) / len(pv); realise = sum(r for _, r i
   n pv) / len(pv)
    remarques = []
    if len({round(p, 2) for p, _ in pv}) <= 2: remarques.append("ancrage : prévisi
   ons quasi identiques")
    if moyenne_annoncee - realise > 0.10: remarques.append("sur-confiance (annoncé
    supérieur au réalisé)")
```

À toi, en solo · réponse

Pistes de correction (suite)

```
    if realise - moyenne_annoncee > 0.10: remarques.append("sous-confiance")
    return remarques

rng = random.Random(8)
def previsions(personne, n=60):
    out = []
    for _ in range(n):
        vrai = rng.uniform(0.15, 0.85); r = 1 if rng.random() < vrai else 0
        if personne == "calibrée": p = vrai
        elif personne == "sur-confiante": p = min(0.98, max(0.02, 0.5 + (vrai - 0.
   5) * 2.4 + 0.15))
        else: p = 0.5
        out.append((p, r))
    return out
for qui in ("calibrée", "sur-confiante", "toujours 50 %"):
    pv = previsions(qui); print(f"{qui:15s} Brier {brier(pv):.3f} | {biais(pv) or
   'aucun biais détecté'}")
b = {q: brier(previsions(q)) for q in ("calibrée", "sur-confiante", "toujours 50 %
   ")}
```

À toi, en solo · réponse

Pistes de correction (suite)

```
assert biais(previsions("sur-confiante")) and "ancrage : prévisions quasi identiqu
   es" in biais(previsions("toujours 50 %")) and b["calibrée"] < b["toujours 50 %"
   ] < b["sur-confiante"] + 0.1
d = Decision("Pivot vers le B2B", ["persévérer", "pivoter", "arrêter"], "pivoter",
    ["VE la plus haute", "revenu concentré sur le B2B"], criteres_preetablis=True,
    pre_mortem=False, couts_irrecuperables_exclus=True)
score, manques = d.processus(); print(f"qualité du processus : {score:.0%} ; manqu
   es : {manques}")
assert manques == ["pré-mortem fait"]
```

La **qualité d'une décision** se juge sur le **processus** (critères préétablis, options, pré-mortem, coûts irrécupérables exclus, raisons écrites) et non sur le résultat seul. La **calibration** se mesure sur des dizaines de prévisions : une personne qui répond toujours 50 % n'est pas sur-confiante, mais elle n'apporte aucune information (score de Brier médiocre) ; une personne sur-confiante annonce des probabilités extrêmes qui ne se réalisent pas.

Pièges classiques

Piège	Pourquoi c'est un problème	Comment l'éviter
Décider d'après le coût déjà dépensé	On engage de l'argent pour sauver le passé	Ne regarder que la valeur espérée de l'avenir
Modifier les critères après avoir vu les résultats	On s'arrange avec les chiffres	Critères relus et appliqués avant l'analyse
Juger une décision à son résultat	On apprend de la chance, pas de la méthode	Évaluer processus, information, options ; tenir un journal
« Encore trois mois » sans critère	Escalade d'engagement	Seuil d'arrêt et horizon écrits d'avance

Mini-boss

Dossier MVP final de HashBox : de l'idée à la décision. Reprends les productions du module (problème, hypothèses, MVP, UX, architecture, priorisation, analytics, dette) pour une idée de produit de ton choix et rassemble un dossier complet, avec code :

1. **cadrage** (mission 1) : problème sans solution, segment, JTBD, étoile du nord, arbre de métriques ;
2. **découverte** (mission 2) : 10 entretiens simulés analysés (qualité, thèmes, saturation), opportunité principale ;
3. **hypothèses et expériences** (mission 3) : 3 hypothèses avec critères, une expérience simulée avec test statistique et règle de décision ;
4. **MVP** (mission 4) : story map, tranche mince, MoSCoW, critères d'acceptation exécutables ;
5. **UX** (mission 5) : entonnoir, contraste, audit d'accessibilité, résultats d'un test de 8 personnes ;
6. **architecture** (mission 6) : matrice robuste, construire/acheter, fonctions d'aptitude ;

Mini-boss (suite)

7. **roadmap** (mission 7) : RICE, robustesse, Smith, plan sous capacité ;
8. **analytics** (mission 8) : plan de mesure, validation, cohortes, un cas de Simpson, NPS ;
9. **delivery** (mission 9) : flux, dette, prévision Monte Carlo ;
10. **revue de décision** (mission 10) : critères préétablis contre résultats avec intervalles, options (persévérer/pivoter/arrêter) par valeur espérée sans coût irrécupérable, valeur de l'information, pré-mortem, biais repérés, **mémo de décision** et 5 prévisions chiffrées ;
11. une **rétrospective** de 20 lignes : ce que le dossier t'a appris, la décision que tu aurais prise sans lui, ce qui aurait changé ton avis.

Mini-boss · réponse

Pistes de correction

Un excellent dossier : cohérence de bout en bout (l'étoile du nord du cadrage est la métrique des critères de la revue), chaque section produit des chiffres réutilisés dans la suivante, le mémo de décision cite les intervalles et exclut le coût irrécupérable, la valeur de l'information justifie (ou non) une expérience supplémentaire, les prévisions chiffrées sont notées pour la calibration future, et la rétrospective admet au moins une conclusion inconfortable (par exemple un biais personnel repéré).

Quiz éclair

Quiz éclair

1. Dans une décision de continuer ou d'arrêter, le coût déjà dépensé :

A. ne doit pas entrer dans la comparaison des options futures
B. est l'argument principal
C. doit être doublé

2. La valeur de l'information parfaite représente :

A. le coût de l'expérience
B. le plafond de ce qu'il vaut la peine de payer pour réduire l'incertitude
C. le gain maximal du produit

3. Une prévision est bien calibrée quand :

A. tous les événements annoncés se réalisent

Quiz éclair

Quiz éclair (suite)

B. les probabilités annoncées sont toujours de 50 %

C. les événements annoncés à 70 % se réalisent environ 70 % du temps

4. 🔁 *Révision mission 09* : La dette technique peut se comparer à une dette financière parce que :
A. elle accélère aujourd'hui mais impose des intérêts (ralentissement) sur chaque modification future

B. elle n'a aucun coût

C. elle se rembourse toute seule

5. 🔁 *Révision mission 09* : Selon la loi de Little, à débit constant, réduire le travail en cours :
A. augmente le temps de cycle

B. réduit le temps de cycle moyen

Quiz éclair

Quiz éclair (suite)

C. n'a aucun effet

Quiz éclair · réponse

Réponses

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

Je réfléchis

1. Quelle décision récente de ton équipe a été défendue par « on a déjà tant investi » ?
2. Si tu devais écrire aujourd'hui le seuil à partir duquel tu arrêterais ton projet actuel, quel serait-il ?

À retenir

À retenir

1. Critères préétablis relus avant les résultats ; verdict avec intervalles.
2. Persévérer, pivoter, arrêter : valeur espérée de l'avenir, coût d'opportunité inclus.
3. Le coût irrécupérable n'entre pas dans la décision.
4. Valeur de l'information : plafond du prix d'une expérience.
5. Pré-mortem, journal de décisions, calibration (Brier), mémo de décision écrit.

Félicitations

Félicitations

Tu as terminé **HEF-150 — Ingénierie produit**. Tu sais transformer un besoin en problème testable, mener une découverte avec des faits, formuler et tester des hypothèses avec rigueur, concevoir un MVP de bout en bout, mesurer l'expérience utilisateur, choisir une architecture adaptée au produit, prioriser avec des critères robustes, instrumenter et lire des données, gérer dette et flux, et prendre une décision de produit informée.

→ Durant le module : **HEF-160 — IA et augmentation humaine**

MISSION ACCOMPLIE

+120 XP

Badge débloqué : Décideur éclairé

Fin du module HEF-150. Bravo !

