
Créer un e-book de A à Z

La méthode ASCALEA Book Factory pour transformer un sujet en livre structuré, visuel, Web et PDF

ASCALEA — MYGASPOZ

Créer un e-book de A à Z

La méthode ASCALEA Book Factory pour transformer un sujet en livre structuré, visuel, Web et PDF

Table des matières

La méthode ASCALEA Book Factory pour transformer un sujet en livre structuré, visuel, Web et PDF

Méthode ASCALEA — Structure · Balance · Impact

Introduction — Un e-book n'est pas un fichier, c'est un système de production

Ce que vous allez apprendre

À retenir

Chapitre 1 — Commencer par le Book Brief, pas par la rédaction

1. Pourquoi le brief vient avant le contenu
2. Ce que fait la Book Factory quand vous donnez seulement un sujet
3. Le workspace d'un nouveau livre
4. Le Gate G0

Exemple concret

À retenir

Chapitre 2 — Construire une source maître et une architecture de contenu

1. Pourquoi une seule source maître
2. Du brief au sommaire
3. Les Content Blocks
4. La composition mixed-media
5. La source et les preuves

Exemple concret

À retenir

Chapitre 3 — CONTENT → DEPTH → INFORMATION DESIGN → READER UX

1. CONTENT : construire le fond
2. DEPTH : un seul livre, plusieurs niveaux
3. INFORMATION DESIGN : rendre la structure visible
4. READER UX : penser le parcours du lecteur
5. Pourquoi ces quatre étapes sont séparées

Exemple concret

À retenir

Chapitre 4 — Transformer le contenu en expérience visuelle

1. Le Book Design System
2. Les rôles typographiques
3. Les composants réutilisables
4. Le Golden Book comme référence

5. A4 et branding

Exemple concret

À retenir

Chapitre 5 — Créer les visuels utiles et les couvertures Premium éditoriales

1. Le rôle des Visual Assets

2. La couverture Premium éditoriale par défaut

3. La règle des deux couches

4. Couverture et 4e de couverture

5. Le Gate G4

Exemple concret

À retenir

Chapitre 6 — Publisher : produire le Web Book et le PDF A4

1. Ce que fait Publisher

2. Les règles d'impression

3. La table des matières

4. Web et PDF : même contenu, comportements différents

5. Le Publisher ne doit pas inventer

Exemple concret

À retenir

Chapitre 7 — Quality Assurance : contrôler sans tout recommencer

1. La première étape est l'analyse d'impact

2. Les hard gates passent avant les scores

3. Les niveaux de défaut

4. Maximum deux cycles automatiques

5. Les outcomes

6. Pourquoi la QA protège aussi les coûts

Exemple concret

À retenir

Chapitre 8 — Comment fonctionne la Book Factory dans GitHub

1. L'ordre de lecture au démarrage

2. Puis viennent les fichiers de méthode

3. La hiérarchie des sources de vérité

4. Le State Model

5. Les triggers

6. Le mode RUN_TO_COMPLETION

7. Comment changer de sujet

Exemple concret

À retenir

Conclusion — Du sujet au livre accepté

À retenir

Plan d'action — Lancer votre prochain e-book

Glossaire

Sources de la méthodologie

La méthode ASCALEA Book Factory pour transformer un sujet en livre structuré, visuel, Web et PDF

Méthode ASCALEA — Structure · Balance · Impact

Introduction — Un e-book n'est pas un fichier, c'est un système de production

Créer un e-book semble simple : choisir un sujet, écrire quelques chapitres, ajouter une couverture et exporter un PDF. C'est précisément cette apparente simplicité qui crée le plus de problèmes. On peut obtenir un document lisible sans obtenir un **système de production fiable**.

La Book Factory ASCALEA part d'une idée différente : un e-book professionnel n'est pas seulement un fichier final. C'est une chaîne contrôlée qui relie une intention, un public, une source maître, des règles éditoriales, un design, des visuels, un moteur de publication et une assurance qualité.

Le point de départ n'est donc pas : « écris-moi un livre ». Le point de départ est : **quel problème le livre doit-il résoudre, pour qui, avec quelle promesse et quelles contraintes ?**

Ensuite, chaque étape transforme le même contenu sans le dupliquer :

Sujet → Book Brief → Book Master → Design System → Visual Assets → Publisher → PDF / Web → QA

Cette logique produit deux bénéfices. Le premier est éditorial : le livre reste cohérent, parce qu'une seule source maître gouverne le contenu. Le second est opérationnel : lorsqu'un défaut est découvert, on corrige uniquement l'élément concerné et ses dépendances, au lieu de tout recommencer.

Ce que vous allez apprendre

Dans ce guide, vous allez comprendre :

- comment démarrer un nouvel e-book sans recopier un ancien projet ;
- pourquoi le `BOOK_BRIEF.md` doit exister avant le manuscrit ;
- comment `BOOK_MASTER.md` devient la source unique ;
- ce que font CONTENT, DEPTH, INFORMATION DESIGN et READER UX ;
- comment le Design System et les Visual Assets interviennent ;
- pourquoi la couverture Premium éditoriale est construite en deux couches ;
- comment Publisher fabrique un Web Book et un PDF A4 ;
- comment la Quality Assurance (QA) ferme le cycle ;

dans quel ordre exact les fichiers GitHub sont lus.



Du sujet au livre accepté : la chaîne gouvernée de la Book Factory ASCALEA.

À retenir

Un e-book professionnel est le résultat d'un **workflow gouverné**. Le PDF n'est que l'une de ses sorties.

Chapitre 1 — Commencer par le Book Brief, pas par la rédaction

1. Pourquoi le brief vient avant le contenu

La première erreur classique consiste à commencer par écrire. Le problème n'est pas l'écriture elle-même ; c'est l'absence de contrat préalable.

Le `BOOK_BRIEF.md` sert de contrat de cadrage. Il fixe au minimum :

- le sujet ;
- le titre et le sous-titre ;
- le problème lecteur ;
- le public cible ;
- le niveau de départ ;
- la profondeur maximale ;
- la promesse ;
- l'objectif du livre ;
- le ton ;
- la langue maître ;
- les exigences de sources ;
- les contraintes ;
- les sorties attendues.

Tant que ces éléments ne sont pas définis, la rédaction peut être élégante mais mal orientée.

2. Ce que fait la Book Factory quand vous donnez seulement un sujet

La méthodologie accepte une requête très courte :

Crée un nouvel e-book sur [SUJET].

Le système ne demande pas automatiquement une série de préférences déjà gouvernées. Il crée un `book_id` , ouvre un workspace indépendant et applique les décisions acceptées.

Les défauts hérités incluent notamment :

- Markdown comme source maître ;
- PDF A4 ;
- nouveau chapitre sur nouvelle page ;
- branding ASCALEA imprimé ;
- table des matières paginée ;
- Option C — Premium éditorial pour les couvertures ;
- Web drill-down et PDF entièrement développé ;
- mode `RUN_TO_COMPLETION` .

3. Le workspace d'un nouveau livre

Chaque nouveau sujet reçoit son propre espace :

```
docs/book-factory/08-books/<book_id>/
```

Le Golden Book ne doit pas être écrasé. Il reste la référence validée, pas le modèle à copier manuellement.

La structure minimale du nouveau livre contient :

```
BOOK_BRIEF.md
TABLE_OF_CONTENTS.md
BOOK_MASTER.md
COVER_VISUAL_BRIEF.md
FULL_BOOK_VISUAL_APPLICATION.md
PUBLISHER_RUN_REPORT.md
FINAL_QA_REPORT.md
visuals/
outputs/
```

4. Le Gate G0

Avant de continuer, le Gate G0 vérifie que le brief est suffisamment complet :

- sujet clair ;

public identifiable ;
promesse définie ;
sorties connues ;
aucune prémisse centrale qui obligerait à inventer.

Si un élément facultatif manque, le workflow peut appliquer un défaut gouverné ou l'omettre. Si un élément central manque réellement, il ne doit pas être fabriqué.

Exemple concret

Vous demandez : « Crée un e-book sur la gestion des risques liés à l'intelligence artificielle pour les PME. »

La Book Factory crée un nouveau `book_id` , puis un Book Brief qui transforme cette phrase en périmètre exploitable : public PME, thème risque IA, objectif pratique, sorties Web/PDF, règles ASCALEA héritées.

À retenir

Le Book Brief réduit l'ambiguïté avant qu'elle ne devienne du contenu inutile.

Chapitre 2 — Construire une source maître et une architecture de contenu

1. Pourquoi une seule source maître

Une règle centrale de la Book Factory est simple : **un contenu maître produit plusieurs sorties**.

Le texte n'est pas réécrit séparément pour le Web et le PDF. Le `BOOK_MASTER.md` devient la source de vérité éditoriale du livre.

Cette règle évite trois dérives :

- une version PDF différente du Web ;
- des corrections appliquées dans un format mais oubliées dans l'autre ;
- des coûts de maintenance inutiles.

2. Du brief au sommaire

Une fois le brief établi, la table des matières organise la promesse du livre en progression logique.

Le sommaire ne sert pas uniquement à afficher des chapitres. Il prépare :

- les identifiants ;
- les ancres ;
- la navigation ;
- les futurs liens internes ;
- la pagination du PDF ;
- le parcours Web.

La hiérarchie canonique est :

```
BOOK
  → CHAPTER
    → SECTION
      → CONTENT BLOCK
```

3. Les Content Blocks

Un bloc peut être :

- du texte ;
- une définition ;
- un exemple ;
- un tableau ;
- un diagramme ;
- une image ;
- un callout ;
- un avertissement ;
- un message à retenir ;
- une note source ;
- un drill-down.

Le choix du média dépend du message, pas d'un quota décoratif.

4. La composition mixed-media

La méthodologie accepte texte, tableaux, graphiques, diagrammes et images quand ils améliorent réellement l'explication.

La règle de prudence est importante :

- un graphique quantitatif exige des données traçables ;
- un diagramme peut expliquer une relation qualitative ;
- une image doit avoir un objectif ;
- le PDF et le Web doivent transmettre le même message.

5. La source et les preuves

Le Book Master ne doit pas inventer ce que les sources ne disent pas.

Si le livre part d'un document utilisateur, celui-ci reste la référence initiale. Si le workflow exige un enrichissement externe, il doit être traçable.

Exemple concret

Dans le Golden Book, aucun graphique chiffré n'a été créé pour les 10 zones à risque parce que le manuscrit ne fournissait aucune série quantitative. Un diagramme qualitatif a été retenu à la place.

À retenir

BOOK_MASTER.md gouverne le contenu publié. Le design transforme sa présentation, pas sa vérité.

Chapitre 3 — CONTENT → DEPTH → INFORMATION DESIGN → READER UX

1. CONTENT : construire le fond

Le pipeline P1 Content vérifie :

- sujet et promesse ;
- logique du contenu ;
- exactitude ;
- clarté ;
- exemples adaptés ;
- explication des termes techniques.

Le résultat attendu est un contenu maître prêt à être structuré.

2. DEPTH : un seul livre, plusieurs niveaux

La méthodologie ne crée pas trois livres séparés pour débutant, intermédiaire et expert.

Elle classe plutôt les blocs :

Niveau	Rôle
Essential	nécessaire pour comprendre et agir
Intermediate	approfondissement utile
Advanced	détail expert lorsque réellement disponible

Sur le Web, l'Intermediate et l'Advanced peuvent être révélés progressivement. Dans le PDF, le contenu est développé.

3. INFORMATION DESIGN : rendre la structure visible

L'Information Design organise :

- titres et sous-titres ;

numérotation ;
ancres ;
table des matières ;
tableaux ;
séquences ;
callouts ;
blocs Exemple concret ;
blocs À retenir.

Son objectif est de réduire la charge cognitive.

4. READER UX : penser le parcours du lecteur

L'expérience utilisateur (UX) du lecteur pose une question simple : **le lecteur sait-il toujours où il est et quoi faire ensuite ?**

Le modèle retenu favorise :

des titres explicites ;
une orientation claire ;
des liens précédent / sommaire / suivant ;
des libellés descriptifs ;
un drill-down utile, pas décoratif.

5. Pourquoi ces quatre étapes sont séparées

Elles travaillent sur des problèmes différents :

Content : qu'est-ce qu'on dit ?
Depth : à quel niveau de profondeur ?
Information Design : comment l'information est-elle organisée ?
Reader UX : comment le lecteur la parcourt-il ?

Les fusionner trop tôt pousse souvent à corriger la mise en page alors que le vrai problème est éditorial.

Exemple concret

Un chapitre très long peut sembler être un problème de design. Après analyse, la vraie correction peut être de reclasser des blocs Intermediate, de créer un tableau de synthèse et de rendre un exemple repliable sur le Web.

À retenir

La mise en page ne doit pas compenser un contenu mal structuré.

Chapitre 4 — Transformer le contenu en expérience visuelle

1. Le Book Design System

Une fois le contenu structuré, le Design System donne au livre une grammaire visuelle réutilisable.

Le baseline accepté définit notamment :

- bleu marine #082E5F ;
- vert #35A547 ;
- rouge ASCALEA #CE1B2E ;
- fond blanc ;
- surfaces très légères ;
- hiérarchie sobre ;
- espaces généreux ;
- tableaux simples ;
- callouts cohérents.

2. Les rôles typographiques

Le système distingue :

- kicker de chapitre ;
- titre de chapitre ;
- promesse lecteur ;
- titre de section ;
- corps ;
- tableau ;
- métadonnées.

La typographie n'est pas un embellissement. Elle matérialise la hiérarchie éditoriale.

3. Les composants réutilisables

Les composants principaux incluent :

chapter opener ;
tableau gouverné ;
EXEMPLE CONCRET ;
À RETENIR ;
checklist ;
figure ;
navigation interne.

L'objectif n'est pas d'avoir beaucoup de composants, mais de réutiliser les mêmes conventions pour que le lecteur apprenne rapidement le langage du livre.

4. Le Golden Book comme référence

Le Design System a été accepté après Publisher et QA du Golden Book. Cela signifie qu'un nouveau livre n'a pas besoin de redécider la couleur des tableaux ou la position des éléments de marque.

Il hérite du baseline, sauf override explicite.

5. A4 et branding

Le PDF canonique utilise :

A4 210 × 297 mm ;
marges fixes ;
wordmark ASCALEA + slogan en haut ;
phoenix en bas à droite ;
copyright en bas à gauche ;
numéro physique en bas à droite.

Chaque nouveau chapitre commence sur une nouvelle page.

Exemple concret

Une nouvelle production peut changer complètement de sujet tout en conservant la même logique visuelle. Le contenu évolue ; la grammaire du livre reste stable.

À retenir

Un Design System accepté transforme les décisions visuelles répétitives en règles réutilisables.

Chapitre 5 — Créer les visuels utiles et les couvertures Premium éditoriales

1. Le rôle des Visual Assets

Le pipeline Visual Assets intervient quand un visuel améliore réellement la compréhension.

Les formats possibles incluent :

- schéma ;
- matrice ;
- graphique ;
- diagramme ;
- infographie ;
- illustration.

Chaque asset doit avoir :

- un objectif ;
- une source ;
- un type ;
- une légende ;
- un texte alternatif ;
- un comportement Web ;
- un comportement PDF ;
- un statut.

2. La couverture Premium éditoriale par défaut

Pour les nouveaux e-books, **Option C — Premium éditorial** est le mode par défaut.

Le système ne demande plus à l'utilisateur de choisir entre plusieurs niveaux visuels si aucune dérogation n'est demandée.

3. La règle des deux couches

La couverture doit séparer :

```
VISUEL SUJET
+
TEXTE ÉDITABLE
+
ASSETS DE MARQUE OFFICIELS
↓
COUVERTURE FINALE
```

Le titre, le sous-titre et le logo officiel ne doivent pas être irréversiblement intégrés dans une image générée.

Pourquoi ? Parce qu'une image peut comporter une faute, un logo déformé ou un texte impossible à modifier.

4. Couverture et 4e de couverture

La couverture combine :

- logo officiel ;
- titre exact ;
- sous-titre ;
- un visuel fort lié au sujet ;
- une promesse claire.

La 4e combine :

- texte source ;
- bénéfices ;
- public cible ;
- un visuel cohérent mais secondaire ;
- signature ASCALEA.

5. Le Gate G4

Le Gate G4 vérifie notamment :

cohérence du Design System ;
pertinence des visuels ;
Option C par défaut ;
séparation image / texte / logo ;
provenance et alt text.

Exemple concret

Si une image générée contient déjà le titre et un faux logo ASCALEA, elle peut servir d'inspiration mais ne doit pas devenir la couche finale. Le Publisher doit conserver le texte et les assets officiels séparément.

À retenir

Le premium vient de la composition globale, pas du fait de figer tout le design dans une seule image.

Chapitre 6 — Publisher : produire le Web Book et le PDF A4

1. Ce que fait Publisher

Publisher transforme le Book Master et les assets en sorties exploitables.

La chaîne PDF canonique est :

```
BOOK_MASTER.md
→ HTML
→ PRINT_A4.css
→ WeasyPrint
→ PDF
→ render verification
→ QA
```

2. Les règles d'impression

Le CSS impose notamment :

- format A4 ;
- marges ;
- nouveau chapitre sur nouvelle page ;
- titres non orphelins ;
- tableaux et figures non coupés quand ils tiennent sur une page ;
- header ASCALEA répété ;
- footer copyright et pagination.

3. La table des matières

Les numéros de page de la table des matières ne sont pas saisis manuellement.

Le moteur utilise la cible du lien pour calculer la page finale :

```
.toc-list a::after {
  content: target-counter(attr(href), page);
}
```

Si la pagination change, les numéros sont régénérés.

4. Web et PDF : même contenu, comportements différents

Le Web Book peut utiliser :

- drill-down ;
- navigation locale ;
- liens profonds ;
- contenus repliables.

Le PDF est statique et complet.

Les deux doivent partager :

- terminologie ;
- titres ;
- design language ;
- contenu source.

5. Le Publisher ne doit pas inventer

Si un asset manque ou si un texte n'existe pas dans le Book Master, Publisher ne doit pas fabriquer une version pour « remplir ».

Il assemble ce qui a été gouverné.

Exemple concret

Dans le Golden Book, un visuel SVG a été rasterisé à haute résolution au moment du PDF parce que le moteur d'impression rendait mal le texte SVG. Le SVG est resté la source canonique ; le PNG n'était qu'un dérivé de rendu.

À retenir

Publisher est un moteur d'assemblage et de rendu, pas un auteur silencieux.

Chapitre 7 — Quality Assurance : contrôler sans tout recommencer

1. La première étape est l'analyse d'impact

La QA ne commence pas en revalidant tout.

Elle commence par :

- identifier ce qui a changé ;
- identifier ce qui n'a pas changé ;
- identifier les dépendances ;
- réutiliser les preuves encore valides ;
- définir les contrôles à rejouer.

2. Les hard gates passent avant les scores

Un score de 95/100 ne peut pas compenser :

- une contradiction avec une décision ACCEPTED ;
- un fait inventé ;
- une source fabriquée ;
- un contenu accepté manquant ;
- un lien critique cassé ;
- un asset nécessaire absent.

3. Les niveaux de défaut

Niveau	Effet
Critical	empêche une publication fiable
Major	altère substantiellement compréhension ou conformité
Minor	défaut réel mais non matériel pour l'usage principal
Observation	point à consigner sans correction immédiate

4. Maximum deux cycles automatiques

Chaque skill peut corriger ses défauts ciblés puis re-scoring, avec un maximum de deux cycles automatiques.

Après cela, la méthodologie préfère :

- simplifier ;
- omettre un bloc non fiable ;
- réduire le périmètre ;
- consigner une exception ;

plutôt qu'inventer.

5. Les outcomes

Les outcomes possibles incluent :

- COMPLETE ;
- COMPLETE_WITH_WARNINGS ;
- COMPLETE_WITH_EXCEPTIONS ;
- HARD_FAIL .

6. Pourquoi la QA protège aussi les coûts

La règle de Validation Reuse évite de rerendre et revérifier 20 pages si seules la couverture et la conclusion ont changé.

Exemple concret

Lors de la finalisation du Golden Book, les couvertures ont été modifiées. Les pages internes validées ont été réutilisées ; seule la zone impactée et la pagination aval ont été retestées.

À retenir

Une bonne QA ne contrôle pas tout à nouveau. Elle contrôle exactement ce que le changement peut avoir cassé.

Chapitre 8 — Comment fonctionne la Book Factory dans GitHub

1. L'ordre de lecture au démarrage

Lorsqu'un nouveau livre est demandé, l'orchestrateur lit :

```
00-governance/CURRENT_GOVERNING_STATE.md
00-governance/DECISIONS.md
01-book-methodology/NEW_BOOK_START_HERE.md
01-book-methodology/BOOK_CREATION_METHOD.md
07-templates/BOOK_BRIEF_TEMPLATE.md
```

Il crée ensuite le `BOOK_BRIEF.md` du nouveau livre.

2. Puis viennent les fichiers de méthode

Après le Brief : 6. `COVER_VISUAL_STRATEGY.md` 7. `P1_CONTENT_PIPELINE.md` 8.
`P2_DEPTH_PIPELINE.md` 9. `P3_INFORMATION_DESIGN_READER_UX_PIPELINE.md` 10.
`P4_EDITORIAL_VISUAL_DESIGN_PIPELINE.md` 11. `P5_VISUAL_ASSETS_PIPELINE.md` 12.
`P6_TECHNICAL_PUBLISHING_PIPELINE.md` 13. `SKILL_QA.md`

P7 Workflow & Orchestration supervise l'ensemble.

3. La hiérarchie des sources de vérité

Si deux fichiers semblent se contredire, la hiérarchie est :

```
GOUVERNANCE ACCEPTED
↓
BOOK BRIEF
↓
BOOK MASTER
↓
DESIGN SYSTEM + COVER STRATEGY
↓
VISUAL ASSETS
↓
PUBLISHER
↓
QA
```

4. Le State Model

Un livre progresse par états :

```
DRAFT
→ SPECIFIED
→ CONTENT_READY
→ DEPTH_READY
→ UX_READY
→ DESIGN_READY
→ VISUALS_READY
→ PUBLISH_READY
→ ACCEPTED
→ PUBLISHED
```

L'état permet à l'orchestrateur de savoir quelle étape exécuter ensuite.

5. Les triggers

Exemples :

CONTENT_READY → P2 Depth ;

DEPTH_READY → P3 ;

UX_READY → P4 ;

DESIGN_READY → P5 si un asset est requis ;

VISUALS_READY → P6 Publisher.

6. Le mode RUN_TO_COMPLETION

Une fois le run lancé :

aucune validation intermédiaire pour les choix déjà gouvernés ;
pas de question esthétique si le défaut existe ;
corrections automatiques ciblées ;
poursuite jusqu'aux sorties finales ou à un vrai hard fail central.

7. Comment changer de sujet

Il suffit de lancer une nouvelle demande.

La Book Factory ne remplace pas l'ancien livre. Elle crée un nouveau workspace sous :

```
08-books/<nouveau-book-id>/
```

Le Golden Book et les autres livres restent intacts.

Exemple concret

Ce livre lui-même a été créé comme nouveau sujet : la méthodologie Book Factory constitue son contenu source, tandis que le Design System et les règles d'impression viennent du baseline accepté.

À retenir

GitHub contient la méthode, l'état, les preuves et les sources du livre. Le lecteur n'a pas besoin de manipuler ces fichiers pour lancer un run.

Conclusion — Du sujet au livre accepté

Une Book Factory utile ne cherche pas à automatiser chaque geste pour le plaisir d'automatiser. Elle cherche à **transformer les décisions répétitives en règles**, tout en conservant des points de contrôle là où la fiabilité l'exige.

Le modèle final repose sur quelques principes simples :

- un nouveau sujet crée un nouveau workspace ;
- le Book Brief définit le contrat ;
- le Book Master gouverne le contenu ;
- les pipelines transforment le contenu sans le dupliquer ;
- le Design System stabilise l'expérience visuelle ;
- les Visual Assets n'existent que s'ils servent le message ;
- Publisher produit les sorties ;
- QA contrôle les hard gates et réutilise les preuves.

La conséquence est importante : vous pouvez changer complètement de sujet sans reconstruire la méthode.

Le prochain e-book commence donc par une phrase très simple :

Crée un nouvel e-book sur [SUJET].

La complexité reste dans la Book Factory. La demande, elle, reste simple.

À retenir

Le vrai actif n'est pas seulement le PDF final. C'est la méthode reproductible qui permet d'en produire un autre avec le même niveau de contrôle.

Plan d'action — Lancer votre prochain e-book

Étape	Action
1	Formulez le nouveau sujet en une phrase
2	Laissez la Book Factory créer le <code>book_id</code> et le workspace
3	Vérifiez que le Book Brief exprime le bon problème lecteur
4	Laissez Content, Depth et Information Design structurer le fond
5	Appliquez le Design System accepté
6	Créez uniquement les visuels réellement utiles
7	Laissez Publisher produire Web + PDF A4
8	Exécutez la QA finale et fermez les défauts
9	Passez le livre à <code>ACCEPTED</code>

Glossaire

Terme	Définition
Book Brief	contrat de cadrage d'un nouveau livre
Book Master	source maître du contenu publié
Book Design System	règles visuelles réutilisables du livre
Content Pipeline	pipeline qui construit et fiabilise le fond
Depth	classification Essential / Intermediate / Advanced
Information Design	organisation visuelle et structurelle de l'information
Reader UX	expérience de lecture et de navigation
Visual Asset	schéma, image, tableau visuel ou graphique gouverné
Publisher	moteur qui transforme la source maître en Web/PDF
QA	Quality Assurance, assurance qualité finale
Hard Gate	condition non compensable par un score
Validation Reuse	réutilisation des preuves encore valides
RUN_TO_COMPLETION	mode autonome sans validation humaine intermédiaire pour les choix gouvernés
Golden Book	livre de référence utilisé pour accepter le système
Book State	état courant d'un livre dans le workflow

Sources de la méthodologie

Ce livre est dérivé de la méthodologie Book Factory actuellement ACCEPTED dans le dépôt mygaspoz , notamment :

01-book-methodology/NEW_BOOK_START_HERE.md
01-book-methodology/BOOK_CREATION_METHOD.md
01-book-methodology/COVER_VISUAL_STRATEGY.md
01-book-methodology/BOOK_CONTENT_MODEL.md
02-orchestrator/ORCHESTRATOR_SPEC.md
02-orchestrator/BOOK_STATE_MODEL.md
02-orchestrator/TRIGGERS_AND_ROUTING.md
03-pipelines/P1_CONTENT_PIPELINE.md à
P7_WORKFLOW_ORCHESTRATION_PIPELINE.md
05-design-system/BOOK_DESIGN_SYSTEM.md
05-design-system/PDF_SPEC.md
06-workflow/AUTONOMOUS_EXECUTION_POLICY.md
06-workflow/END_TO_END_WORKFLOW.md
06-workflow/QUALITY_GATES.md
06-workflow/QUALITY_SCORING_FRAMEWORK.md
08-golden-book/FINAL_QA_REPORT.md

La méthodologie décrit le processus réel utilisé pour produire et valider les e-books de la Book Factory.