# なぜブログを作ったのか、なぜAstroなのか

「いそしぎ忘備録」の技術選定背景

2026-05-18

# 完璧主義でうまくいかない問題

- 当初は理由のわからないまま、公開や発信が一向に進まなかった
- アウトプットしようとすると強い抵抗を感じ、気づけば作業が停止している
- 完璧を求めすぎているのではないかという漠然とした自覚はあったが、原因を具体的に言語化できなかった
- **結果**：中間成果物は滞留し、知見は抽出されず、行動そのものが停止した

# 完璧主義の正体を自己分析する

- なぜ公開できないのかを掘り下げた結果、一貫した基準が浮かび上がった
- 「今考える限り最も価値が高い状態」を満たさない成果物は公開に値しないと判断していた
- 失敗が判明した中間成果物は「最善の価値」を持たないとみなされ、公開が躊躇される
- 「将来失敗が判明するかもしれない」という不確実性だけでも公開が停止する

# さらに掘り下げると、価値提供への歪んだこだわりだった

- 「最も価値が高い状態」とは、突き詰めれば**「最大の価値を提供しなければならない」という歪んだこだわり**
- 完璧でなければ価値がないと思い込み、公開そのものを止めてしまっていた
- しかし、価値提供の本質は「完璧」ではなく「届ける」こと
- 公開しなければ、どんな価値も誰にも届かない

# だから、価値提供のプラットフォームを作るしかなかった

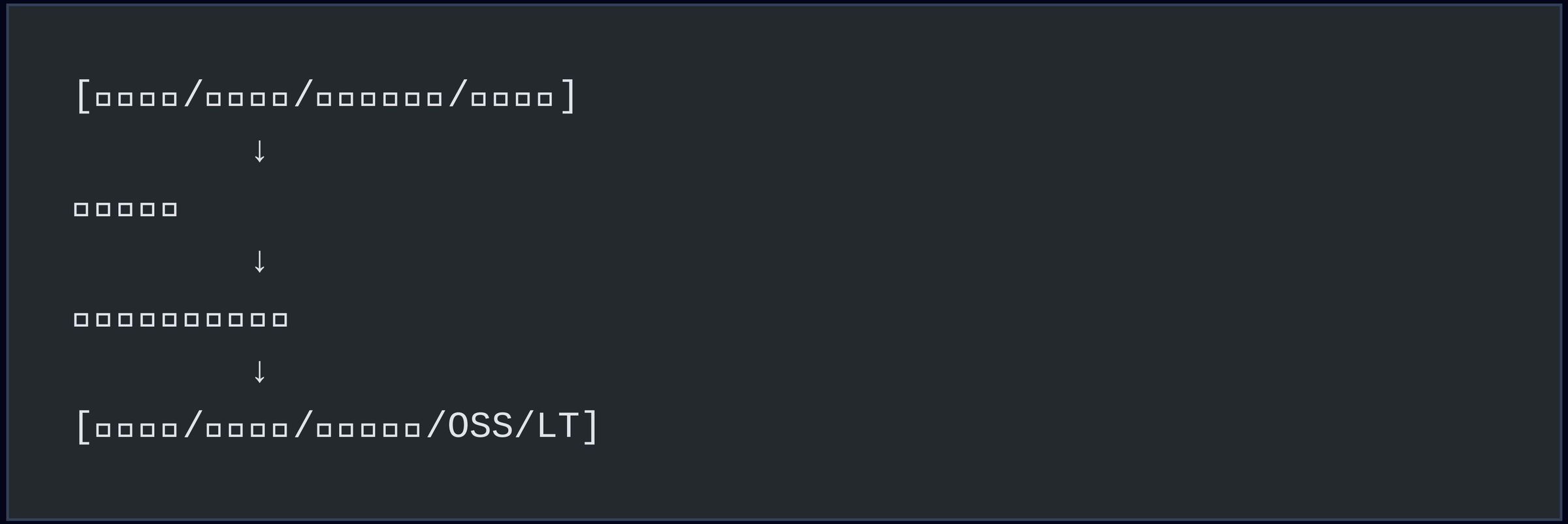
- 公開しなければ価値は届かない → **まず公開する仕組み**が必要
- 既存プラットフォーム（Zenn, Qiita）は記事投稿に最適化されており、多様なアウトプットに向かない
- 自分専用のプラットフォームで、**記事・スライド・ストーリー・ログなど多様なフォーマット**を同一基盤で蓄積する
- 自分のペースで、中間成果物からの抽出→物語化→公開のサイクルを回す土台

# 解決の核心：物語として完成させる

- 完璧主義そのものは否定しない。高い品質を追求する姿勢は尊重されるべき
- 問題は**行動が停止すること**
- 中間成果物から抽出した知見を**物語として完成させる**ことで、価値提供の要求を満たす

- 失敗は物語化によって資産化され、回収される
- つまり、完璧を求める感情は実は副次的な要素に過ぎなかった
- 完璧でなくとも、価値提供が可能であればサイクルは回せる

# 中間成果物から小さな成果への循環



[□ 証結果 / 試行記録 / 未公開コード / 活動記録 ]

↓

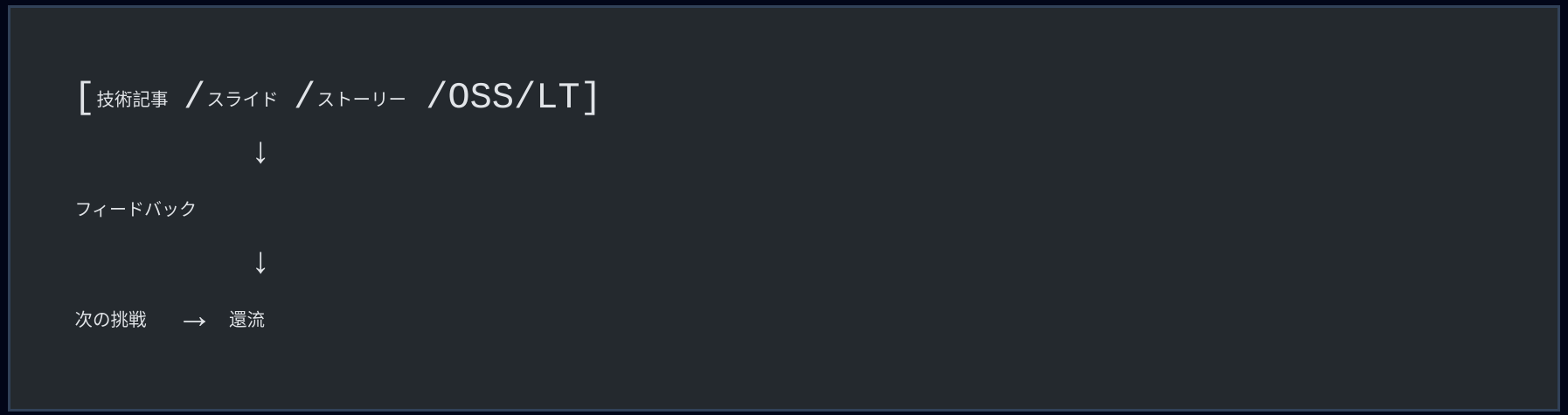
知見を抽出

↓

物語として完成させる

↓

[技術記事 / スライド / ストーリー /OSS/LT]


[技術記事 /スライド /ストーリー /OSS/LT]
↓
フィードバック
↓
次の挑戦 → 還流

- 中間成果物そのものの完成度を求めてはならない
- 抽出された知見の物語化にこそ価値がある
- 循環を回す仕組みを持つこと自体が、完璧主義による行き詰まりを防ぐ

# 5つの運用規則

1. **物語として完成させ、資産化し回収する** — 中間成果物そのものを公開対象としない
2. **回収を前提に試行を増やす** — 失敗しても物語として回収できる前提があれば試行コストは相殺される
3. **アウトプットの規模を調整する** — 時間が不足する場合、成果の規模を縮小してよい

4. **品質へのこだわりの再発に対処する** — 規則1に即座に立ち返る
5. **内部の蓄積を判断基準とする** — 外部反応がない期間は過去の自己との比較に切り替える

# 演劇論的アプローチ

- 規則を設けても、意志の力だけではいずれ枯渇する
- だから意志に依存せず、パイプラインを駆動し続けるための「演劇論」を導入する
- 「価値を提供する者」としての役割を定義し、それを演じ続ける
- 気分の有無にかかわらず、小さな成果を抽出する行動が導かれる
- **役割としての振る舞いを選択し続けることが、サイクルの自動化となる**

# 技術選定の指針：やらないことを決める

| やらないこと | 技術選定への影響 |
| --- | --- |
| CMS的な編集・管理機能 | SSG一択。ビルドして公開するだけ |
| 重いアニメーション・複雑なインタラクション | ゼロJSデフォルトのフレームワーク |

| やらないこと | 技術選定への影響 |
| --- | --- |
| 他者向けコメント・SNS連携機能 | シンプルな静的出力、外部依存を減らす |
| SEO最適化を完璧に仕上げる | 公開停止を防ぐため、過度な最適化を諦める |

| やらないこと | 技術選定への影響 |
| --- | --- |
| 完璧なデザインシステム | シンプルなユーティリティCSS |
| 汎用的なツール化 | 自分専用。汎用性より簡素さを優先 |
| コストをかける | 無料・低コストのホスティング |

# なぜAstroか (1)：動的要素ゼロ、完全SSGの必然性

- 「CMS不要」「複雑なインタラクション不要」→ 動的な要素が一切存在しない
- SSRやCSRのランタイムを動かす理由がなく、完全SSGこそが唯一の合理的選択
- Astroのコア思想である **“Zero JS by default”** と、自分の「やらないこと」が完全に一致

- デプロイも単なるファイルアップロード
  - ビルドアーティファクトを投げるだけ。エッジでのコンピューティング（SSR）すら捨てる潔さ
  - `dist/` をWorker Static Assetsに置くだけ。NodeサーバーもDBも不要
  - CDNで世界中に高速配信
  - 無料枠内で十分運用可能

# なぜAstroか (2)：MDXで統一された執筆体験

- **記事もスライドもMDX** で同じフォーマット
  - CMS的な管理画面を排除し、ファイルベースの執筆に徹する
  - テンプレートをコピーするだけで書き始められる
  - 認知負荷を下げ、公開のハードルを下げる

- **Content Collections + Zod スキーマ**
  - `content.config.ts` でフロントマターの構造を定義
  - ビルド時にバリデーションが走り、型安全性を保証
  - `description` の200文字制限、`tags` の重複除去、`draft` による公開除外

- **MDX 内に任意のコンポーネントを埋め込み可能**
  - `<SlideCard slideTitle={frontmatter.title}>` コンポーネントやカスタムUIをマークダウン内に直接配置
  - 記事・スライド・ストーリーで同じコンポーネントセットを再利用

- 同じフォーマットで多様な表現ができ、自分のペースで蓄積できる

# なぜAstroか (3)：ビルド後の自動化

- ビルド済みの静的HTMLにローカルサーバーを立て、Playwright を起動
- クライアントJSゼロの完全な静的サイトだから、全ページの巡回と処理がシンプルに完結する
- OGP画像とスライドPDFを自動生成するビルド後処理を実装

- **OGP画像の自動生成**（Playwright）
  - 各記事・スライド・ストーリーのOGPをビルド後に自動作成
  - 画像を手作りするコストが公開のハードルを上げるのを防ぐ

- **スライドのPDF出力**（Playwright）
  - スライドをPDFとして自動出力
  - 同じフォーマットで蓄積した知見を別の場面でも再利用可能

- **書くだけで完結** — ビルドすれば画像もPDFも生成済み

# スタイリング選定：Tailwind CSS v4

- **CSS-first 構成：** `tailwind.config.js` を廃止
- `@theme` とCSS変数でテーマを `src/styles/global.css` 内に閉じて定義
- ビルド時に未使用スタイルを除去し、バンドルサイズを最小化

- **色の制約：sky + slate のみ**
  - 2色軸に絞ることでデザイン判断のコストを限界まで下げる
  - 「完璧なデザインシステムはやらない」という制約を実装レベルで満たす

# ホスティング選定：Cloudflare Worker Static Assets

- **Worker Static Assets** はPagesの後継。静的アセットをデプロイする新方式
- **Cloudflareのエッジネットワーク** からグローバル配信
- `npx wrangler deploy` で dist/ 全体を一括デプロイ
- 無料枠で運用可能。Analytics は Cloudflare エッジレベルで自動注入

# コンテンツの2層設計

- **通常コンテンツ**（articles / slides / stories）：共通スキーマ + MDX本文。物語として完成させた成果
- **軽量コンテンツ**（events）：フロントマターのみ、個別ページなし。少しでも多くのアウトプットを担保するための機能

# 全体アーキテクチャ

```
src/content/
├── articles/   → /articles/[slug]    技術記事
├── slides/     → /slides/[slug]      スライド（ + PDF自動出力）
├── stories/    → /stories/[slug]     ストーリー
└── events/     → /logs 一覧のみ          イベント参加記録
```

```
Astro SSG
  ├── MDX
  ├── Tailwind CSS v4 sky + slate
  └── Playwright OGP + PDF
        ↓ npm run build
    HTML dist/
        ↓ npx wrangler deploy
    Cloudflare Worker Static Assets
```

# まとめ

- 完璧主義を否定せず、**仕組みで行動を維持する**
- 中間成果物から知見を抽出し、**物語として完成させる**ことでサイクルを回す

- 技術は **「やらないこと」から逆算して選ぶ**
    - 公開のハードルを下げるための一貫した方針（SSG / MDX / シンプルなスタイリング / 無料ホスティング）

- **小さく公開し、回収し、次に進む** — サイクルを止めずに回し続けること

# Future Work

- **軽量コンテンツの再定義：events から scrap へ**
  - 運用する中で、現行の events には課題が見えてきた
  - 手軽さを求めた結果、情報として軽すぎて評価に困る状態に陥った
  - 次の一手：events を廃止し、「アウトプットの軽さ」と「情報としての価値」を両立させる **scrap** 機能を実装予定
  - 抽出のサイクルをさらに高速に回せるプラットフォームへと進化させる