

PRODUCTION & OPERATIONS

# サイト制作・公開・運用保守 ロードマップ

---

受注後、ローカルHTMLを「使える公開サイト」にし、  
運用保守までを誰でも同じ品質で回すための実務手順。

第1章 制作～公開 (F0-F7) / 第2章 システム構成と画面 / 第3章 運用保守

# 目次

## 01

### 第1章 制作から公開まで

受注準備 → 制作 (F0-F3) → ドメイン・公開 (F4-F7)

## 02

### 第2章 システム構成と画面例

公開サイトの仕組み・完成サイト・月次レポートの実例

## 03

### 第3章 運用保守

落とさない／育てる (分析) ／報告する + 障害・契約

# 採用方式と全体フロー

## 方式: 静的サイト + 外部サービス埋め込み

- ✓ 速い テンプレにJSONを流すだけで土台が完成
- ✓ 安い ホスティングほぼ無料・サーバ管理不要
- ✓ 壊れにくい WordPressの更新事故・乗っ取りが無い
- ✓ 動的機能 予約/問い合わせは実績ある外部SaaSを埋め込む

## 全体フロー

- 受注・キックオフ
- ヒアリング / 素材回収
- 制作 (content.json → build)
- 動的機能を組み込み (フォーム/予約/解析)
- ドメイン取得 → デプロイ公開
- QA / SEO / GBP連携
- 納品・レクチャー・請求
- ▶ ▼ 運用保守フェーズへ

## CHAPTER 1 ・ BUILD & PUBLISH

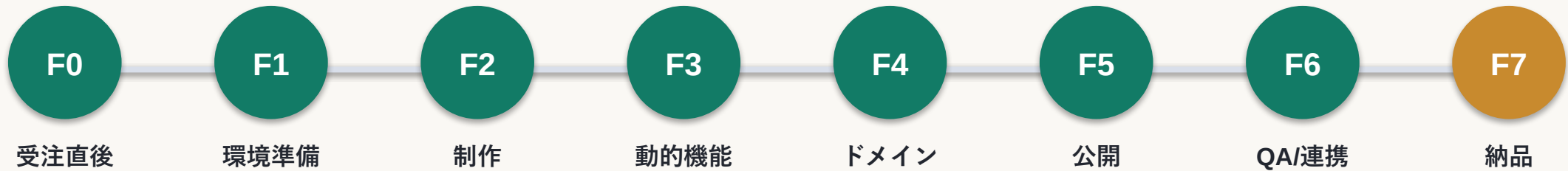
### 第1章

# 制作から公開まで

---

- 受注直後の準備とヒアリング (F0)
- 制作環境・テンプレートでサイト化 (F1-F2)
- 動的機能を動かす：フォーム・予約・計測 (F3)
- ドメイン取得・デプロイ・QA・納品 (F4-F7)

# 制作～公開の7フェーズ



**標準リードタイム:** 素材がそろってから公開まで最短5～10営業日（スタンダード）。

最大の関門は F3 「動的機能を動かす」 = 予約・問い合わせを実際に届く状態にすること。

# 受注直後の準備

受注直後

ゴール: 売る土台（情報と段取り）を固める

キックオフ日程を確定（30分ヒアリング枠）

ヒアリングシートを一緒に埋める

素材回収の段取り（写真・ロゴ・原稿・SNS）

ドメイン / メール方針を確認

公開希望日を確定（繁忙期の前を狙う）

## ここで決める3点

- 使うテンプレート
- 予約の方式（電話/予約SaaS/フォーム）
- ドメインは新規 or 既存

## 制作環境・ツール一式（一度だけ整える）

用途	サービス	費用
コード/履歴管理	GitHub	無料
ホスティング	Cloudflare Pages（推奨） / Netlify / Vercel	無料
ドメイン	Cloudflare Registrar / お名前.com	~1,500円/年
問い合わせフォーム	Tally / Formspree	無料枠
ネット予約	Airリザーブ / RESERVA / 飲食はebica等	無料~
アクセス解析	GA4 + Microsoft Clarity	無料
画像最適化	Squoosh / TinyPNG	無料

ローカルは Python3 だけあればよい（build\_site.py 実行用）。このリポジトリ構成をそのまま使う。

## 制作（ローカルで形にする）

制作

ゴール: content.json + 写真で土台を完成

ヒアリング内容を content.json に転記

写真を圧縮（長辺1600px・各150KB以下）

原稿を整える（AI叩き台→店の言葉に）

build\_site.py でビルド

スマホ幅で崩れないか確認

### 品質の8割は写真

- 撮影 or 既存を厳選
- images/に同梱しURLは相対パス
- URL切れ防止・表示も高速

## 動的機能を「動く」状態にする

### 問い合わせフォーム

Tally 埋め込み or Formspree。  
HTMLフォーム送信→店のメール/LINEへ。  
reCAPTCHA/ハニーポットでスパム対策。

### ネット予約

飲食=ebica/トレタ、サロン=Airリザーブ等。  
予約ページをリンク/iframe埋め込み。  
作り込まない店は電話+フォームで充分。

### 計測 (必須)

GA4 + Microsoft Clarity を全サイトに。  
フォーム送信・電話タップをイベント計測。  
→ 月次レポートの数字になる。

電話 tel: と Googleマップ iframe は即動く。 フォーム・予約・計測を入れて初めて『使えるサイト』になる。

# ドメイン取得・DNS

## ドメイン

ゴール: サイトの住所を用意する

ドメイン名を決定 (短く・店名・.com/.jp)

レジストラで取得 (Cloudflareは原価更新)

店名義で取得し管理代行 = 店の資産 (推奨)

既存ドメインはネームサーバ/DNSを向ける

既存サイト/メールを止めない切替手順を共有

## 名義の選択

- 店名義 (資産・推奨)
- 自社名義 → 解約時に移管
- 契約書の移管条項に従う

# デプロイ（公開） — Cloudflare Pages

1

## リポジトリ準備

sites/<店名> をGitHubに置く

2

## Pages連携

Connect to Git → リポジトリ選択

3

## ビルド設定

Build command: なし / 出力: sites/<店名>

4

## 仮URL確認

xxxx.pages.dev で表示チェック

5

## 独自ドメイン

Custom domains で追加→DNS設定

6

## SSL自動

証明書自動発行→ https 表示でOK

Git不要なら “Upload assets” にフォルダをドラッグでも可。既存レンタルサーバがある店はFTPアップロード。

# 公開前後のチェック（QA / SEO / GBP）

## 公開前チェック

- ☑ スマホ/PCで表示崩れなし
- ☑ 電話・地図・予約・問い合わせが実際に動く(テスト送信)
- ☑ 全ページ https（混在警告なし）
- ☑ 店名/住所/電話/営業時間が正確・GBPと一致
- ☑ 画像が重すぎない（表示が速い）
- ☑ 誤字・価格ミスなし（店に最終確認）
- ☑ title/description/OGP・favicon設定

## SEO・GBP連携

- title/descにエリア + 業種 + 店名
- 構造化データ(LocalBusiness)
- Search Console登録 + サイトマップ
- GBPのサイト/予約リンクを更新
- GBPの写真・営業時間を最新化

# 納品 ・ レクチャー ・ 請求

納品

ゴール: 公開して終わりにしない

納品物・管理情報を整理 (URL/各アカウント)

操作レクチャー (届く場所・更新依頼の仕方)

アカウント/パスワードを管理台帳に記録

初期費の残金を請求・月額サブスク開始

保守内容・窓口・対応時間を書面で渡す

渡す / 保持を分ける

- 店に渡す情報
- こちらが管理する情報
- 台帳で一元管理

第2章

# システム構成と画面例

---

- 公開サイトの仕組み（静的+外部SaaS）
- 完成サイトの画面例（スマートフォン）
- 月次レポートの画面例（顧客ポータル）

# システム構成図（公開サイトの仕組み）



## 制作・反映



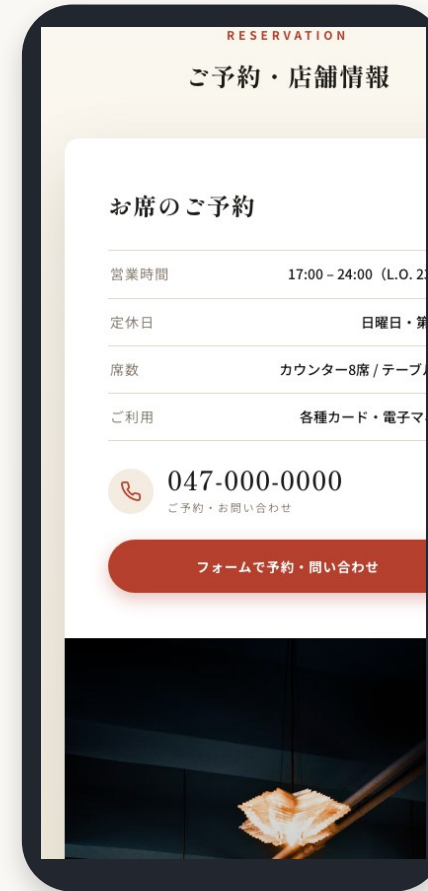
# 完成サイトの画面例（スマートフォン）



トップページ



お品書き・写真



ご予約・お問い合わせ

テンプレートに店舗データを流し込むだけで、スマホ最適化された同品質のサイトを量産できる。

## 月次レポートの画面例（毎月、顧客ポータルで公開）



KPI（前月比）・Googleの口コミ/アクセス状況・所感を1枚に。この提出が“払う価値”の可視化＝解約防止の要。

## 第3章

# 運用保守ロードマップ

月額サービスの中身 = 3本柱

01

落とさない

死活監視・バックアップ・障害対応・  
ドメイン期限管理

02

育てる

口コミ自動返信・月次レポート

03

報告する

月次レポートで成果を可視化 = 解約防  
止の最大の武器

属人化が事故になる → 顧客管理台帳を最初から作る（ドメイン期限・各SaaS・請求状況）。

## 日常運用 と 月次運用

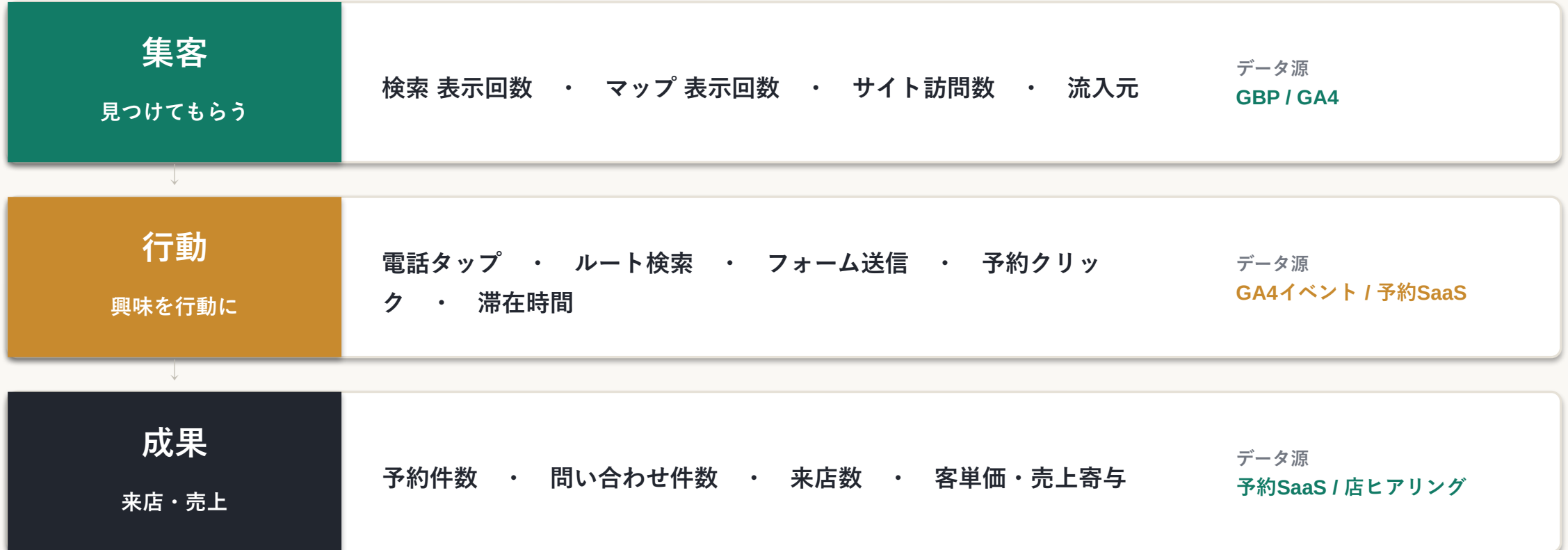
### 日常運用（随時）

- 死活監視（UptimeRobot等）で落ちたら通知
- 更新依頼: content.json編集→再ビルド→再デプロイ
- バックアップ: Gitに全履歴（画像も同梱）
- 月1でフォーム/予約のテスト送信
- 対応SLAを決める（軽微な更新は2営業日以内）

### 月次運用（毎月の定例）

- GBP/GA4/Clarityの数字を取得
- 口コミにAIで返信（承認制）
- 月次レポート生成→顧客ポータルに公開
- 気づきをレポートに添える

## 効果測定的设计 — 何を計測するか



**前提:** 公開時に GA4 / Clarity のイベント計測（フォーム送信・電話タップ）を仕込んでおく。仕込まないと“成果”が測れない。

## 数字をどう読むか（分析の観点）

### トレンドで見る

前月比・前年同月比で増減と季節性を把握。  
点でなく線で判断。

### 流入元の内訳

検索 / マップ / SNS / 直接 の比率。どこを  
伸ばすか決める。

### 行動率（CVR）

問い合わせ ÷ 訪問。サイトの“効き”。低け  
れば導線を改善。

### ヒートマップ・離脱

Clarityで見られない箇所・離脱点を発見。  
改善の根拠に。

### 検索キーワード

どんな語で来たか → 訴求・文言・メ  
ニュー名を最適化。

### 変化に気づく

前月との差分から、店の手応えや変化を読  
み取る。

## 月次レポートで“価値”を可視化する

### レポートの構成（章立て）

- 01 ご挨拶・今月のサマリー
- 02 今月のKPI（前月比つき）
- 03 Google・アクセスの状況
- 04 今月の口コミ対応
- 05 所感・気づき

### 伝え方の原則

- ✓ 専門用語を使わない（数字を日本語で）
- ✓ 「予約〇件増」など店の利益で語る
- ✓ 良い点も課題も正直に書く
- ✓ 気づいた点を一言添える
- ✓ 毎月同じ日に提出してリズムを作る
- ✓ 対面 or 電話で5分、口頭でも説明する

04\_サイトテンプレート/report で自動生成 → 毎月の公開が“払う価値”の証明＝解約防止の最大の武器。

## 数字を見て・気づいて・伝えるサイクル



🔄 毎月この1周を回す。数字の確認（見る）と報告（伝える）はセット。レポートで成果を可視化する。

続けるほど信頼が積み上がり、解約が減り、紹介が生まれる → MRRと事業価値が積み上がる。

### 第3章

## 障害・トラブル対応フロー

事象	一次対応	恒久対応
サイトが表示されない	監視通知で即検知→DNS/ホスティング確認	原因切り分け・再デプロイ
ドメイン期限切れ（最重要）	自動更新ON徹底・期限を台帳管理	期限60日前にアラート
SSLエラー	証明書の自動更新状況を確認	ホスティング側で再発行
フォーム/予約が届かない	テスト送信で再現→通知先/迷惑メール確認	サービス設定見直し
スパム大量	reCAPTCHA/ハニーポット強化	提供元の対策ON
炎上口コミ相談	冷静な返信文・事実確認	改善アクションを一緒に検討

ドメイン期限切れだけは絶対防ぐ（復旧不能・信用失墜）。自動更新ON＋台帳＋カレンダー通知の三重で守る。

## 更新・アップセル・解約 / 推奨構成

### 契約の回し方

**更新** 更新月前に成果サマリー提示→自然に継続

**アップセル** ページ追加/LP制作/予約導線の強化（都度の制作）

**解約・移管** 残債案内→データ/ドメイン移管→GBP返却→停止

**KPI** 稼働率/対応速度/レポート提出率/継続率/紹介数

### 推奨スターター構成

- ホスティング: Cloudflare Pages
- ドメイン: Cloudflare Registrar
- 問い合わせ: Tally
- 予約: Airリザーブ（飲食はebica等）
- 解析: GA4 + Microsoft Clarity
- 監視: UptimeRobot
- 会計/決済: freee/MF + Stripe/Square

→ 1サイトの原価はドメイン代（年~1,500円）のみ

# まとめ

静的サイト + 外部SaaS埋め込みで、速く・安く・壊れにくく作る。

F3『動的機能を動かす』が関門。フォーム・予約・計測まで入れて初めて使えるサイト。

公開はCloudflare Pages（無料・独自ドメイン・SSL自動）。原価はドメイン代のみ。

運用保守 = 落とさない・育てる・報告する。月次レポートが解約防止の要。

ドメイン期限切れは三重で防止。属人化は台帳で潰す。

詳細手順は [docs/サイト制作・公開・運用保守ロードマップ.md](#) / 作業はチェックリストで管理。