- Clarify purpose, audience, and constraints, then summarize recommendations
- Inventory all projects/components and score them against a clear rubric
- Recommend Keep / Refactor / Archive actions with lean scopes and estimates
- Build a 90-day cleanup roadmap with weekly checkpoints for a 2-person team
- Identify top quick wins to unlock progress immediately
- Define risks, metrics, and a cadence to keep momentum and accountability

Project Inventory & Decision Map — StackStudio Ecosystem

TL;DR

This document consolidates every active and proposed project in the StackStudio ecosystem, applies a consistent decision rubric, and sets a 90-day cleanup roadmap for a two-person team. The intent is to stop thrashing, focus on the highest-leverage assets (StackFast/StackStudio core), and retire or defer distractions while creating a clear path to revenue.

Goals

Business Goals

- Ship a revenue-capable StackFast path (trial-to-paid) within 60–90 days.
- Reduce maintenance overhead by 40–60% by archiving low-value projects and refactoring critical debt.
- Increase deployment frequency to at least 2x/week with stable CI/CD.
- Establish a single-source-of-truth inventory and roadmap with weekly status updates.

User Goals

- Offer a fast, reliable path for builders to launch sites/apps (StackFast) with minimal setup.
- Provide power users a stable customization surface (StackStudio) backed by a coherent design system and templates.
- Deliver trustworthy docs and starter templates that reduce time-to-first-value to under 15 minutes.

Non-Goals

- Launching speculative R&D (e.g., net-new AI concepts) in the next 90 days.
- Supporting broad third-party integration surface beyond the 2–3 essential ones.
- Building a large team or complex processes; operate lean with two people.

User Stories

- As a Founder/PM, I want a single inventory with clear recommendations, so that I can make fast, confident decisions.
- As an Engineer, I want a prioritized, scoped backlog, so that I can deliver reliably without context switching.

- As an Early Customer, I want a simple way to sign up, build, and pay, so that I can launch quickly and trust the platform.
- As a Support/Success Owner, I want essential docs and telemetry, so that I can help users without guessing.

Functional Requirements

- Project Inventory & Scoring (Priority: P0)
  - Inventory table capturing: name, owner, status, user value, technical debt, maintenance cost, dependencies, recommended action, priority.
  - Filtering by priority/action; export as CSV/Notion.
  - One-click archive flag with linked repo/asset references.
- Decision Rubric (Priority: P0)
  - Clear thresholds for Keep / Refactor / Archive based on user value, revenue potential, maintenance cost, and feasibility.
  - Evidence fields (usage, leads, support volume) to justify scores.
- Recommendations Engine (Priority: P0)
  - Per-project action with rationale, scope, and estimated effort (in person-days) for a two-person team.
  - Owner assignment and due-by window within the 90-day plan.
- Roadmap Planner (Priority: P0)
  - 90-day phases with weekly checkpoints, deliverables, and demo criteria.
  - WIP limits and “no parallel P0” rule to reduce thrash.
- Metrics & Tracking (Priority: P1)
  - Define success metrics and instrument top user flows (signup, build, publish, pay).
  - Weekly rollup (deploy frequency, cycle time, open criticals, MQLs, paid).
- Documentation & Appendix (Priority: P1)
  - Source inventory list with short summaries.
  - Decision log capturing what changed and why.

Project Inventory

Name	Owner	Status	User Value	Technical Debt	Maintenance Cost	Dependencies	Recommended Action	Priority
StackFast (user-facing builder)	Martin (PM)	Alpha	High: core revenue path	Medium	Medium (~8h/mo)	Templates, Auth/Billing, API, DevOps	Keep; harden path to payment	P0
StackStudio (power-user studio)	Engineer	Prototype	Medium: upsell for advanced users	High	High (~16h/mo)	StackFast core, Design System, API	Refactor: scope to essentials	P1
Unbuilt (AI scaffolding)	Martin	Paused/Concept	Low: speculative in 90 days	Medium	Medium (~6h/mo if maintained)	LLM APIs, Templates	Archive/Icebox for 90 days	P2

Templates Library	Engineer	Beta	High: accelerates delivery	Medium	Medium (~6h/mo)	Design System	Refactor: consolidate + version v1.0	P0
Design System (UI kit)	Martin + Engineer	In-progress	High: consistency + velocity	Medium	Low (~3h/mo)	—	Keep; ship DS v1	P0
API Gateway/Backend	Engineer	Live	High: platform backbone	Medium	Medium (~8h/mo)	Auth, Database	Keep; security + reliability hardening	P0
Auth & Billing (Stripe/OAuth)	Engineer	Live (basic)	High: enables monetization	High	Medium (~6h/mo)	Stripe, OAuth provider	Refactor: stabilize + edges	P0
Integrations (3rd-party)	Martin	Mixed	Medium: only a few matter	High	High (~10h/mo)	Third-party APIs	Refactor: keep Stripe/Zapier; archive low-use	P1
Docs & Tutorials	Martin	Outdated	Medium: reduces support	High	Medium (~4h/mo)	Product features	Refactor: publish essentials	P1
Marketing Site & Branding	Martin	Live	High: lead-gen	Low	Low (~2h/mo)	Design assets	Keep; update messaging to single narrative	P0
DevOps/CI/CD	Engineer	Live	High: stability + speed	Medium	Low (~3h/mo)	Repos	Keep; standardize pipelines	P0
Analytics & Telemetry	Engineer	Partial	Medium: informs focus	Medium	Low (~2h/mo)	Frontend/Backend	Refactor: instrument core flows	P1

User Experience

Entry Point & First-Time User Experience

- Access: Open the Inventory board (this document’s table or companion tracker).
- Onboarding: Add or confirm each project row; assign an owner; rate user value, debt, and maintenance using provided definitions.

Core Experience

- Step 1: Score projects
  - Use the decision rubric definitions below each field.
  - Validate with any available evidence (usage stats, support pings).
  - Mark quick obvious Archives immediately.
- Step 2: Apply recommendations
  - For each Keep/Refactor, confirm scope and estimate (in person-days).
  - Assign owner and target week within 90 days.
- Step 3: Build the roadmap
  - Drag P0 items into Phase 1–2; enforce no parallel P0s.
  - Add weekly demo criteria per item (what proves it's done).
- Step 4: Execute weekly cadence
  - Monday: plan; Friday: demo; keep decision log updated.
  - Update the inventory and burndown metrics.
- Step 5: Review outcomes
  - Evaluate success metrics monthly; adjust priorities.

Advanced Features & Edge Cases

- Power users: filter by action/priority; export a "kill list" for archives.
- Error states: conflicting dependencies; flag and resolve via decision log.
- Edge: emergent opportunity; gate via a 1-page spike with 2-day timebox.

UI/UX Highlights

- Use clear status tags (Alpha/Live/Paused/Archived).
- Color-code priorities (P0 red, P1 amber, P2 gray).
- Keep forms minimal; default options for speed.
- Accessibility: consistent typography; high contrast; responsive layout.

Narrative

The team is small—two people with a pile of projects and no clear line to revenue. Prospects are confused by overlapping brands and an inconsistent experience. Each week brings new ideas, yet the backlog grows and confidence stalls. The goal is to stop the spin and focus on what matters.

This decision map starts by listing every asset—StackFast, StackStudio, templates, backend, docs—and scoring them against one rubric. Overnight the noise becomes signal: StackFast plus a clean design system and payments is the shortest path to value. StackStudio is promising, but it needs guardrails. A few integrations and a speculative AI concept can wait.

The plan imposes a 90-day rhythm. Week by week, the team archives dead weight, refactors core components, and ships a stable trial-to-paid flow. The marketing site tells a single story. Docs finally match the product. Telemetry shows what customers actually do, guiding each next step. By the end, shipping is normal, maintenance is down, and there's an honest shot at revenue—built by a team of two moving fast and focused.

Success Metrics

User-Centric Metrics

- Time-to-first-value: < 15 minutes from signup to publish.
- Activation rate: ≥ 40% of signups complete a first publish.
- NPS or CSAT on onboarding: ≥ 40.
- Churn proxy (7-day retention for early users): ≥ 30%.

Business Metrics

- Trial-to-paid conversion: ≥ 5–8% within 90 days.
- Qualified pipeline: 20 demo requests or 50 MQLs by Day 90.
- Revenue: 1–3 paid pilots or first $1–3k MRR by Day 90.

Technical Metrics

- Deployment frequency: ≥ 2/week sustained.
- Critical bug backlog: -70% by Day 60.
- MTTR: < 24 hours; error rate < 1% on core flows.
- Test coverage on core modules (auth, payment, publish): ≥ 60%.

Tracking Plan

- Events: signup_started, signup_completed, project_created, template_selected, publish_success, payment_started, payment_success, error_auth, error_publish.
- Operational: deploy_count_weekly, MTTR, open_criticals, cycle_time.
- Growth: pageviews_marketing, MQL_submitted, demo_booked, conversion_paid.

Technical Considerations

Technical Needs

- Data model for Projects (fields in the inventory table), Decisions (action, rationale, date), and Roadmap (phase, owner, due).
- Simple front-end table with filters; export to CSV/Notion; link to repos.
- Back-end CRUD for inventory; audit trail for decision log.

Integration Points

- Repos/CI for status badges; Stripe for billing metrics; basic analytics SDK for product events.
- Optional: Notion/Airtable sync if a shared workspace is preferred.

Data Storage & Privacy

- Store only operational metadata (no PII beyond owner names/emails).
- Follow least-privilege access to third-party APIs and billing data.

Scalability & Performance

- Lightweight needs; prioritize reliability and simplicity over scale.
- Ensure rollups render quickly (< 500ms for typical inventory sizes).

Potential Challenges

- Hidden dependencies causing rework: mitigate with dependency field and weekly reviews.
- Integration flakiness: wrap external calls; monitor and fall back.
- Over-customization: enforce WIP limits and a strict scope template.

Milestones & Sequencing

Project Estimate

Large: 8–12 weeks (aligned to a 90-day plan including buffer and GTM).

Team Size & Composition

Small Team: 2 total people

- Martin (Founder/PM/Design): prioritization, messaging, docs, QA.
- Engineer (Full-stack): implementation, CI/CD, instrumentation.

Suggested Phases

- Phase 0 — Kickoff & Baseline (Days 1–3)
  - Key Deliverables: inventory finalized; rubric adopted; KPIs baseline.
  - Dependencies: access to repos, analytics, Stripe.
- Phase 1 — Decide & Archive (Weeks 1–2)
  - Key Deliverables: archive/icebox list executed; P0/P1 backlog locked; roadmap published.
  - Dependencies: owner confirmation; backup of repos/assets.
- Phase 2 — Core Refactors to Monetization (Weeks 3–6)
  - Key Deliverables: DS v1; Templates v1.0; Auth/Billing hardened; API hardening; publish flow stable; marketing messaging updated.
  - Dependencies: Stripe keys; domain/DNS; CI/CD.
- Phase 3 — Beta, Telemetry, and GTM (Weeks 7–10)
  - Key Deliverables: instrumentation of core flows; docs essentials; first cohort onboarding; pricing/checkout live.
  - Dependencies: analytics SDK; docs hosting.
- Phase 4 — Stabilize, Iterate, Prove (Weeks 11–13)
  - Key Deliverables: fix list burn-down; 2+ deployments/week; demos/pilots; monthly metrics review and next 90-day plan.
  - Dependencies: user feedback loop; support channel.

Weekly Checkpoints (repeat each week)

- Monday: plan sprint, confirm WIP and demo criteria.
- Midweek: remove blockers; adjust only if P0 risk.
- Friday: demo shipped value; update inventory, metrics, and decision log.

Decision Criteria

- User Value
  - High: Directly enables core use case or monetization; validated interest or usage.
  - Medium: Supports adoption/activation; indirect revenue contributor.
  - Low: Nice-to-have; unvalidated or speculative.

- Revenue Potential (90-day lens)
    - Near-term: can unlock/impact revenue in ≤ 90 days.
    - Mid-term: contributes in 3–6 months.
    - Speculative: > 6 months or unclear path.
- Maintenance Cost (owner-hours/month)
    - Low: < 4h; Medium: 4–12h; High: > 12h.
- Technical Feasibility (two-person team)
    - Keep: minimal work; Refactor: ≤ 10–20 person-days; Archive: ≤ 1–2 days to retire safely.
- Recommendation Mapping
    - Keep: User Value High AND (Maintenance ≤ Medium OR Platform-critical).
    - Refactor: User Value High/Medium AND Debt or Maintenance High BUT feasible in ≤ 20 person-days.
    - Archive: User Value Low OR Speculative AND Maintenance ≥ Medium OR lacks owner/clear dependency.

Recommendations

- StackFast — Keep; harden path to payment
    - Scope: stabilize publish flow; add pricing and Stripe checkout; basic telemetry.
    - Estimate: 10–15 person-days.
- StackStudio — Refactor (scope to essentials)
    - Scope: define MVP surface (theme editing, template overrides); cut advanced plugins; align to Design System v1.
    - Estimate: 12–18 person-days.
- Unbuilt — Archive/Icebox
    - Scope: document concept; close branches; mark repo archived; revisit post-revenue.
    - Estimate: 0.5–1 day.
- Templates Library — Refactor
    - Scope: consolidate duplicates; semantic versioning; 80/20 starter set; tests for critical templates.
    - Estimate: 8–12 person-days.
- Design System — Keep; ship v1
    - Scope: tokens, typography, buttons, forms, layout grid; Figma + code alignment; accessibility pass.
    - Estimate: 6–8 person-days.
- API Gateway/Backend — Keep; harden
    - Scope: auth guards, rate limits, logging; publish/persist endpoints; error handling.
    - Estimate: 6–10 person-days.
- Auth & Billing — Refactor
    - Scope: signup/login edge cases; Stripe checkout + webhooks; receipts; downgrade/cancel.
    - Estimate: 8–12 person-days.
- Integrations — Refactor (rationalize)
    - Scope: keep Stripe + Zapier; archive low-use; add “integration policy” (3 proof points before adding).

    - Estimate: 3–5 person-days.
- Docs & Tutorials — Refactor (essentials)
    - Scope: quickstart, pricing, publish, troubleshooting; one video walkthrough.
    - Estimate: 4–6 person-days.
- Marketing Site — Keep; update messaging
    - Scope: unify brand narrative (StackFast primary, StackStudio advanced); clear CTA to trial; case snippets.
    - Estimate: 2–3 person-days.
- DevOps/CI/CD — Keep; standardize
    - Scope: reusable pipeline template; preview deploys; automated checks; rollback playbook.
    - Estimate: 2–4 person-days.
- Analytics & Telemetry — Refactor (core flows)
    - Scope: events for signup, create, publish, pay; dashboards for activation and funnel.
    - Estimate: 3–5 person-days.

Quick Wins (Top 3)

- Centralize the inventory and decision rubric in one place; archive 3 obvious repos today.
- Publish a one-page 90-day roadmap with P0/P1 only; enforce “no parallel P0s.”
- Implement Stripe checkout and a basic pricing page to validate willingness-to-pay.

Risks & Mitigations

- Scope creep and context switching
    - Mitigation: freeze scope weekly; WIP limit of 1 P0 each; change log with explicit trade-offs.
- Hidden technical debt in auth/publish
    - Mitigation: day-1 spike; add error budgets; instrument failures early.
- Revenue runway and signal risk
    - Mitigation: ship pricing/checkout by Week 6; pursue 2–3 paid pilots; weekly outreach cadence tied to demos.
- Integration instability
    - Mitigation: narrow to must-haves; contract tests; disable non-critical integrations by default.

Implementation Notes & Next Steps

Execution Approach

- Operate in weekly cycles: plan Monday, demo Friday, update inventory and metrics.
- Keep a visible decision log with rationale, owner, and date.
- Enforce the rubric; avoid ad hoc additions without a 1-page spike and timebox.

Communication Cadence

- Weekly 30-minute planning; 30-minute demo/retro.
- Daily async check-in (three bullets: yesterday, today, blockers).

Weekly Review Checklist

- Inventory updated; actions accurate.
- Metrics refreshed (deploys, bugs, activation).
- P0s on track; next week's demo criteria defined.
- Decision log entries for any scope change.
- Archive/cleanup tasks executed.

Immediate Next Steps (Week 1)

- Finalize inventory scores and owners; confirm recommendations.
- Execute archive list (Unbuilt, low-use integrations).
- Lock Phase 1 backlog; start Auth/Billing and Templates v1 work.

Appendix: Source Inventory

- Brand and Positioning Notes ("Scribbling it on a napkin") — Summarizes dual-brand approach: StackFast (user-facing) and StackStudio (power users/customization).
- Product Summaries — Current state snapshots for StackFast, StackStudio, and Unbuilt with goals and gaps.
- Templates & Design System Notes — Component lists, naming conventions, and gaps blocking DS v1.
- Auth & Billing Setup — Stripe configuration notes, webhook checklist, and edge cases.
- Integrations List — Inventory of external integrations with usage notes and maintenance costs.
- DevOps/CI/CD — Pipeline templates and deployment checklist; current repos linked.
- Analytics Plan — Proposed event taxonomy and dashboards for activation and funnel.