REFINERY

Where Prose Becomes Perfect

FUNCTIONAL DESIGN DOCUMENT

MVP Specification — All Three Product Lines

Indie Author | Academic | Enterprise Publishing

Version 1.0 — February 2026

CONFIDENTIAL — FOR DEVELOPMENT USE ONLY

DOCUMENT PURPOSE & SCOPE

This Functional Design Document (FDD) defines the complete functional specification for the Refinery MVP across all three product lines — Indie Author, Academic, and Enterprise Publishing. It translates the requirements established in the Requirements Definition Document (RDD v1.0) into actionable development specifications: screen-by-screen UI behavior, API contracts, data models, processing logic, state management, error handling, and sprint-level build sequencing.

This document is the primary reference for the development team throughout the MVP build cycle. It defines what gets built, how it behaves, and in what order. It does not specify implementation language choices or infrastructure tooling beyond what is already established in the tech stack (Flask/FastAPI backend, React frontend, Claude API, Neo4j graph layer).

Section	Contents	Primary Audience
1. System Architecture	High-level architecture, data flow, API structure, authentication	Full stack, DevOps
2. Shared Infrastructure	File ingestion pipeline, AI processing engine, data models, storage	Backend
3. Indie Author MVP	Full screen specs, user flows, module logic for Indie Free + Pro	Frontend, Backend
4. Academic MVP	Full screen specs, user flows, advisor dashboard, academic modules	Frontend, Backend
5. Enterprise Publishing MVP	Triage dashboard, batch processing, multi-editor workflow, API layer	Backend, DevOps
6. Sprint Plan	12-week build sequence, sprint goals, acceptance criteria per sprint	All
7. Testing & QA	Unit test requirements, integration tests, beta acceptance criteria	QA, Backend
8. MVP Exit Criteria	Definition of Done for each product line; pitchable demo checklist	Founders, All

SECTION 1 — SYSTEM ARCHITECTURE

1.1 High-Level Architecture

Refinery is a multi-tenant SaaS application with a shared backend infrastructure serving three distinct frontend experiences. The architecture is designed for the MVP phase to minimize infrastructure complexity while providing the performance and security characteristics required for a pitchable demo and beta user deployment.

Layer	Technology	Role
Frontend	React 18 + TypeScript + Tailwind CSS	Three distinct UX surfaces (Indie, Academic, Enterprise) served from the same
API Gateway	FastAPI (Python 3.11)	Single REST API serving all three products; handles auth, rate limiting, request
Processing Engine	Python async workers (Celery + Redis)	Asynchronous manuscript processing jobs; prevents blocking on long-running A
AI Layer	Claude API (claude-sonnet-4-6)	All manuscript analysis; structured prompt templates per module; long-context c
Manuscript Graph	Neo4j (Community Edition for MVP)	Character relationship graph, scene cross-reference, timeline node map; enable
Primary Database	PostgreSQL 15	User accounts, manuscript metadata, analysis results, version history, audit log
File Storage	AWS S3 (or local MinIO for dev)	Raw manuscript files, processed text, exported documents
Cache / Queue	Redis 7	Job queues for async processing; session cache; rate limit counters
Auth	JWT + refresh tokens; OAuth 2.0 (Google, Microsoft) for Academic/Enterprise	Stateless auth for API; SAML 2.0 for institutional SSO path

1.2 Data Flow — Manuscript Analysis Request

The following sequence describes the complete data flow from manuscript upload through analysis delivery:

- **Step 1 — Upload:** User uploads file via React frontend. Frontend validates file type and size client-side (max 5MB for MVP, ~120K words). File sent as multipart form to POST /api/v1/manuscripts.
- **Step 2 — Ingest:** API gateway receives file, writes raw bytes to S3/MinIO, creates Manuscript record in PostgreSQL with status=PENDING, returns manuscript_id to client.
- **Step 3 — Text Extraction:** Celery worker picks up ingestion job. python-docx or pdfplumber extracts clean text. Smart quotes normalized. Chapter boundaries detected via heading detection heuristics. Extracted text written to S3 alongside raw file.
- **Step 4 — Graph Build:** Second worker runs entity extraction pass via Claude API: extracts character names, scene locations, time markers. Builds Neo4j graph: (Character)-[:APPEARS_IN]->(Scene)-[:PART_OF]->(Chapter). Updates Manuscript status=GRAPH_READY.
- **Step 5 — Module Analysis:** Depending on user-requested modules, one or more analysis workers are queued. Each module fires a structured Claude API call with the full manuscript text plus module-specific prompt template. Results stored as JSON in PostgreSQL analysis_results table keyed by (manuscript_id, module_id, run_id).
- **Step 6 — Delivery:** Frontend polls GET /api/v1/manuscripts/{id}/status via 5-second interval. Once status=COMPLETE, frontend fetches module results from GET /api/v1/manuscripts/{id}/analysis/{module}. Dashboard populates with scores and findings.

1.3 API Structure

All endpoints versioned under /api/v1/. Authentication via Bearer token in Authorization header. All responses return JSON with standard envelope:

```
{ "success": true, "data": {...}, "error": null, "meta": { "request_id": "...", "timestamp": "..." } }
```

Endpoint	Method	Auth	Description
POST /auth/register	POST	None	Create new user account; returns JWT + refresh token
POST /auth/login	POST	None	Email/password login; returns JWT + refresh token
POST /auth/refresh	POST	Refresh token	Issue new JWT from valid refresh token
GET /users/me	GET	JWT	Return authenticated user profile and tier
POST /manuscripts	POST	JWT	Upload new manuscript; returns manuscript_id
GET /manuscripts	GET	JWT	List all manuscripts for authenticated user
GET /manuscripts/{id}	GET	JWT	Get manuscript metadata and current processing status
GET /manuscripts/{id}/status	GET	JWT	Polling endpoint; returns status + % complete
POST /manuscripts/{id}/analyze	POST	JWT	Trigger analysis for specified modules (array of module_ids)
GET /manuscripts/{id}/analysis/{module}	GET	JWT	Fetch analysis results for a specific module
GET /manuscripts/{id}/dashboard	GET	JWT	Fetch aggregated health dashboard for manuscript
POST /manuscripts/{id}/export	POST	JWT	Trigger export job; returns export_id
GET /exports/{export_id}	GET	JWT	Download exported file (DOCX or PDF)
GET /admin/manuscripts	GET	Admin JWT	Enterprise: list all manuscripts in org account
POST /webhooks/submission	POST	API Key	Enterprise: receive submission via webhook

1.4 Authentication & Multi-Tenancy

- Each user belongs to one Account. Accounts have a tier (FREE, INDIE_PRO, ACADEMIC_INDIVIDUAL, ACADEMIC_ADVISOR, ENTERPRISE).
- Feature flags are resolved at the API level based on account.tier — frontend does not enforce access control.
- Enterprise accounts have an additional org_id field enabling multi-user access to shared manuscript pool.
- Academic Advisor accounts have a supervised_users[] array; advisors can read (but not write) any supervised user's manuscripts.
- All PostgreSQL queries include user_id or org_id scope filter — no row-level security gaps permitted.

SECTION 2 — SHARED INFRASTRUCTURE

2.1 File Ingestion & Text Extraction Pipeline

Supported Input Formats (MVP)

- .docx — primary format; processed via python-docx library
- .txt — plain text; chapter detection via blank line + capitalized header heuristic
- .pdf — processed via pdfplumber; text-based PDFs only (no OCR in MVP)
- .rtf — converted to .txt via striprtf library before processing

Chapter Detection Logic

Chapter boundaries are identified via the following priority order:

- Priority 1: Explicit heading styles in .docx (Heading 1, Heading 2 paragraph styles)
- Priority 2: Lines matching regex pattern: ^(Chapter|CHAPTER|Ch\.?|Part|PART)\s+(\d+|[IVXLC]+)
- Priority 3: Lines that are ALL CAPS, standalone (no adjacent text within 2 lines), and shorter than 60 characters
- Priority 4: Three or more consecutive blank lines followed by a short capitalised line
- Fallback: If no chapters detected, treat entire manuscript as one chapter. Flag to user.

Text Normalization

- Smart quotes converted to straight quotes for internal processing; restored on export
- Em dashes (—) and en dashes (–) preserved with surrounding space normalization
- Multiple consecutive spaces collapsed to single space
- Non-printing characters (BOM, zero-width spaces) stripped
- Encoding: normalize to UTF-8 on ingest

2.2 Data Models

Core PostgreSQL Tables

Table	Key Columns	Notes
users	id, email, name, account_id, created_at, last_login	One user per account for MVP (multi-user via org_id for Enterprise)
accounts	id, tier, org_id, stripe_subscription_id, analysis_quota_used, quota_reset_at	Tier: FREE|INDIE|PRO|ACADEMIC_IND|ACADEMIC_ADV|EN
manuscripts	id, user_id, org_id, title, word_count, chapter_count, status, raw_text_s3_key, created_at, updated_at	Status: PENDING|EXTRACTING|READY|ANALYZING|
analysis_runs	id, manuscript_id, module_id, status, started_at, completed_at, token_count, cost_usd	Track API usage per analysis run for cost monitoring
analysis_results	id, analysis_run_id, manuscript_id, module_id, result_json, version, created_at	Full JSON result blob per module per run; versioned
edit_queue_items	id, manuscript_id, module_id, finding_type, chapter_num, location, severity, suggestion, status, created_at	Status: PENDING|ACCEPTED|REJECTED|DEFERRED
manuscript_versions	id, manuscript_id, parent_version_id, label, text_s3_key, word_count, created_at	S3 key per version; enables branching (Arc A / Arc B)
exports	id, manuscript_id, user_id, format, status, output_s3_key, created_at	Format: DOCX|CLEAN_DOCX|TRACKED|PDF_REPORT

Table	Key Columns	Notes
advisor_assignments	id, advisor_user_id, student_user_id, created_at	Academic tier only

2.3 AI Processing Engine — Prompt Architecture

Each analysis module is implemented as a structured prompt template. All prompts follow a consistent pattern to ensure reliable JSON output:

- **System Prompt:** Defines the AI's role as a professional developmental editor, specifies the exact output JSON schema, and sets strict instructions to return ONLY valid JSON with no prose preamble.
- **Context Block:** The full manuscript text, preceded by metadata (title, word count, chapter list, character census from previous graph-build pass).
- **Task Block:** Module-specific instructions (e.g., 'Analyze the dialogue of each character...') with explicit field definitions.
- **Output Schema:** Inline JSON schema specification in the prompt to constrain the model's response format.

Token Management Strategy

- Manuscripts up to ~120K words (~160K tokens) fit within Claude's context window in a single call.
- For manuscripts exceeding this, implement sliding window chunking: overlap 2,000 tokens between chunks, merge results with deduplication.
- Module results are cached: if a manuscript version has not changed since the last run, return cached result without re-calling the API.
- Cost tracking: every API call logs token count and estimated cost to analysis_runs table. Alert threshold: $0.50/single manuscript analysis.

2.4 Neo4j Graph Schema

The manuscript knowledge graph enables cross-chapter querying (e.g., 'find all scenes where Character A and Character B appear together') that cannot be efficiently done with SQL.

Node Type	Properties	Relationships
Manuscript	id, title, word_count	(m)-[:HAS_CHAPTER]->(c)
Chapter	number, title, word_count, start_char, end_char	(c)-[:CONTAINS]->(s), (c)-[:PRECEDES]->(c2)
Scene	id, chapter_num, start_char, end_char, beat_type (ACTION|EMOTION|TRANSITION)	(s)-[:FEATURES]->(chr)
Character	name, aliases[], first_appearance_chapter, frequency_count	(chr)-[:SPEAKS_IN]->(s), (chr)-[:INTERACTS_WITH]->(chr2)
Dialogue	text, speaker_id, chapter_num, word_count	(d)-[:SPOKEN_BY]->(chr), (d)-[:IN_SCENE]->(s)
Metaphor	family, instance_text, chapter_num, frequency	(m)-[:APPEARS_IN]->(c)

SECTION 3 — INDIE AUTHOR MVP

Refinery Free + Pro | Target: Alpha in 12 weeks, Beta-ready in 20 weeks

3.1 MVP Scope Definition

The Indie Author MVP is the foundational product build. It is the primary demo surface for investor pitches, the Kickstarter campaign, and SALT Savannah. The MVP must be functionally complete for the core analysis loop: upload a manuscript, receive a health dashboard, explore at least two modules in depth, and export a result. Full six-module depth is a stretch goal for MVP; core four modules (Intelligence Engine, Prose Refinery, Voice Isolation, Pacing Architect) must be fully functional.

Module	MVP Scope	Post-MVP
Manuscript Intelligence Engine	FULL — duplication detection, character census, lexical fingerprint, health dashboard	Timeline anomaly detection, metaphor heatmap
Prose Refinery	FULL — tic tracker, filter word detector, show vs. tell flags	Sentence rhythm profiler, metaphor frequency manager
Voice Isolation Lab	CORE — dialogue extraction, voice fingerprint, jargon bleed; Blind Test feature post-MVP	Blind Read Test, similarity score index
Pacing Architect	CORE — chapter pulse graph, tension curve, breathing space flags	Scene resequencing sandbox
Character Arc Workshop	BETA ONLY — alternative arc generation is complex; ship basic arc tracking for beta	Alternative arc generation, decision tree
Revision Command Center	PARTIAL — edit queue aggregation and individual accept/reject; batch implementation post-MVP	Batch implementation, version branching, regression detection

3.2 Screen Specifications — Indie Author

Screen 1: Landing / Onboarding

Element	Behavior / Content
Hero	Full-width: 'Meet Refinery. Your manuscript's first honest reader.' CTA: 'Upload Your Draft — It's Free'
Social proof strip	3 beta testimonials (placeholder for MVP launch: 'Beta Author, Romance Fiction' etc.)
Feature preview	3-column: Manuscript X-Ray | Voice Check | Pacing Map — each with short animated GIF or static illustration
Pricing toggle	Free vs. Pro comparison table; 'Start Free' and 'Go Pro — $29/mo' CTAs
Footer	Privacy Policy, Terms of Service, Contact, 'For Publishers' link (leads to Enterprise waitlist)

Screen 2: Upload / New Manuscript

Element	Behavior / Content
File drop zone	Drag-and-drop or click-to-select. Accepted formats shown: .docx .txt .pdf .rtf. Max 5MB shown. Client-side validation fi
Title field	Text input. Auto-populated from filename (minus extension); user can edit. Required.
Genre selector	Dropdown: Literary Fiction | Romance | Science Fiction | Fantasy | Mystery/Thriller | Horror | Historical Fiction | Narrati

Element	Behavior / Content
Upload button	Disabled until file + title present. On click: shows inline progress bar; POST /api/v1/manuscripts. On success: redirect
Free tier notice	Inline banner for FREE users: 'You have 3 free analyses this month. Upgrade to Pro for unlimited.' Hidden for Pro use

Screen 3: Processing / Analysis Progress

Element	Behavior / Content
Progress ring	Animated circular progress indicator. Shows current stage label: 'Reading your manuscript...' → 'Building character ma
Stage timeline	Horizontal stepper: Extract → Map → Analyze → Ready. Current stage highlighted.
Estimated time	Dynamic: 'About 60 seconds remaining.' Updates every 5 seconds from polling response.
Cancel button	Available until status=ANALYZING. On cancel: DELETE /api/v1/manuscripts/{id}/analysis; return to Upload screen.
Error state	If status=ERROR: show error card with message and 'Try Again' button. Log error to Sentry.

Screen 4: Manuscript Health Dashboard

The Dashboard is the primary home screen for a manuscript. It aggregates all module results into a single view.

Component	Behavior / Content
Manuscript header	Title, word count, chapter count, genre, upload date. Edit title inline. Delete manuscript (confirm modal).
Health Score Ring	Large circular score 0–100 in center of page. Color-coded: 0-49 red, 50-74 amber, 75-100 green. Score is weighte
Five dimension scores	Row of 5 score badges: Structure | Voice | Pacing | Character | Prose. Each badge shows score and a 1-line summ
Priority flags panel	Ranked list of top 5 issues across all modules. Each flag shows: module icon, severity badge, short description, ch
Module cards grid	2x3 grid of module cards. Each card: module name, icon, score, status (Complete / Partial / Locked for Free tier), '
Quick stats bar	Word count | Chapter count | Character count | Est. reading time | Last analyzed date
Export button	Prominent CTA: 'Export Annotated Manuscript' — triggers export modal (see Screen 8).
Re-analyze button	Re-runs all modules. Available if manuscript has been updated or 7+ days since last run. Costs 1 analysis credit fo

Screen 5: Module View — Manuscript Intelligence Engine

Component	Behavior / Content
Duplication panel	Table: Chapter A | Chapter B | Similarity % | Preview button. Sort by similarity %. Click Preview opens side-by-side
Character census table	Character name | First appearance (chapter) | Last appearance | Total occurrences | Dialogue lines. Sortable. Clic
Character Profile drawer	Lists every chapter the character appears in. Shows their top 10 distinctive dialogue words. Shows scenes they sh
Lexical fingerprint	Tag cloud of top 50 distinctive words sized by frequency. Toggle: 'All text' vs. 'Dialogue only' vs. 'Narration only'.
Health summary card	Auto-generated prose summary of findings: 'Your manuscript features 14 named characters. Chapter 3 and Chapt

Screen 6: Module View — Prose Refinery

Component	Behavior / Content
Tic tracker	Bar chart: top 20 recurring words, bars colored by chapter-frequency variance. Click word: chapter-by-chapter freq
Filter word panel	Segmented control: All | Seemed | Appeared | Felt | Noticed | Realized | Wondered | (Other). Count per chapter dis
Show vs. tell flags	List view: each flag shows the flagged sentence, the emotion being told, and a suggested show alternative genera
Prose score gauge	Semi-circular gauge 0–100. Below gauge: 3 key metrics shown as stat boxes: Filter Words/1K words | Avg Senten

Screen 7: Module View — Voice Isolation Lab

Component	Behavior / Content
Character selector tabs	Tab bar listing all named characters with dialogue. Click tab: loads that character's voice panel.
Dialogue stream	Scrollable list of all dialogue lines for selected character, in chapter order. Each line shows: chapter number, conte
Voice fingerprint card	For selected character: Avg sentence length | Top 10 unique words | Formality score (0-10) | Jargon category if de
Jargon bleed flags	If jargon detected in a character who should not have it: red flag card with character name, flagged passage, jargo
Compare view	Side-by-side panel: select two characters and see their voice fingerprints compared. Similarity score between the t

Screen 8: Module View — Pacing Architect

Component	Behavior / Content
Pulse graph	Area chart: X-axis = chapters (numbered), Y-axis = 0-10 intensity. Two overlapping areas: Action (blue) and Emoti
Tension curve overlay	Line chart overlaid on pulse graph: cumulative tension score per chapter. Shows rising/falling arc across full manu
Breathing space flags	Below chart: orange warning cards for each 3+ consecutive high-action chapter sequence without emotional decor
Slow zone flags	Red warning cards for each 3+ consecutive low-action chapter sequence without active threat escalation. Specifie
Chapter summary list	Below flags: scrollable list of all chapters, each showing: chapter number, title, dominant beat type (action/emotion

Screen 9: Revision Command Center (MVP — Partial)

Component	Behavior / Content
Edit queue	Paginated list of all findings from all completed modules. Each item: severity icon (High/Med/Low) | module badge
Filter bar	Filter by: Module | Severity | Status (Pending/Accepted/Deferred/Dismissed) | Chapter range.
Queue stats	Top of page: total findings count, breakdown by severity, % addressed.
Item detail drawer	Click 'View': opens right-side drawer with full finding description, AI suggestion, original text snippet, and action bu
Accept action	Marks item as ACCEPTED in edit_queue_items table. Item moves to 'Accepted' tab. Does not auto-edit manuscrip
Export prompt	Persistent CTA at page bottom: 'Ready to revise? Export your annotated manuscript with all findings marked.'

Screen 10: Export

Component	Behavior / Content
Format selector	Radio buttons: Clean DOCX (all accepted changes incorporated) | Tracked Changes DOCX (findings as Word com
Include options	Checkboxes: Include health dashboard summary | Include module scores | Include full findings list | Include sugge
Generate button	POST /api/v1/manuscripts/{id}/export with options. Shows progress spinner. On complete: Download button appea
Pro gate	Tracked Changes DOCX and Analysis Report PDF are Pro-only features. Free users see locked state with 'Upgra

3.3 State Management & Frontend Architecture

- State management: Zustand (lightweight, TypeScript-friendly) for global manuscript and user state.
- Server state: TanStack Query (React Query) for all API calls — handles caching, polling, and loading states.
- Polling: TanStack Query's refetchInterval set to 5000ms on analysis status endpoint; stops polling when status=COMPLETE or status=ERROR.
- Chart library: Recharts for pulse graph and tension curve (React-native, responsive, customizable).
- Error boundary: Each module view wrapped in React ErrorBoundary; module errors do not crash the full dashboard.
- Code splitting: Each module view is lazy-loaded via React.lazy() to minimize initial bundle size.

3.4 Indie Author MVP — Feature Build Status

Feature / Function	Status	Priority	Sprint	Dev Notes
User registration & JWT auth	Build	P0	S1	Standard JWT flow; email verification optional for MVP
File upload + S3 storage	Build	P0	S1	Multipart form; file size validation client + server
Text extraction pipeline (.docx)	Build	P0	S1	python-docx; chapter detection heuristics
Text extraction (.txt, .pdf, .rtf)	Build	P1	S2	pdfplumber for PDF; striprtf for RTF
Character entity extraction (Neo4j build)	Build	P0	S2	Claude API pass; graph write to Neo4j
Manuscript Intelligence Engine — full	Build	P0	S2-3	Duplication, census, fingerprint
Prose Refinery — full	Build	P0	S3	Tic tracker, filter words, show vs. tell
Voice Isolation Lab — core	Build	P0	S4	Dialogue extraction, fingerprint, jargon bleed
Pacing Architect — core	Build	P0	S4	Pulse graph, tension curve, flags
Health Dashboard — aggregated view	Build	P0	S5	Score aggregation from completed modules
Edit Queue — individual accept/reject	Build	P1	S5	DB persistence; no batch in MVP
Export — DOCX clean	Build	P1	S6	python-docx write with accepted edits
Export — DOCX tracked changes	Build	P1	S6	Pro feature; findings as Word comments
Stripe payment integration	Build	P0	S6	Checkout + webhook for subscription status

Feature / Function	Status	Priority	Sprint	Dev Notes
Character Arc Workshop — basic tracking	Build	P2	S7+	Ship at beta; arc tracking only
Voice Blind Test	Defer	P2	Post-MVP	UX complexity; defer to V1 launch
Scene resequencing sandbox	Defer	P2	Post-MVP	Defer to V1 launch
Version branching	Defer	P2	Post-MVP	Arc A/B branching deferred to V1.5

SECTION 4 — ACADEMIC MVP

Refinery Academic | Individual + Advisor tiers; Institutional pilot-ready at Beta

4.1 MVP Scope Definition

The Academic MVP is built on top of the Indie infrastructure with four key additions: the Argument Coherence Engine (replacing Pacing Architect), the Citation & Source Architecture module, Academic Voice Calibration, and the Advisor Dashboard. The institutional SSO path (SAML/LTI) is deferred to V1.5. The Academic MVP must be demo-ready for the Columbus State and Chattahoochee pilot conversations.

Feature	MVP Scope	Defer to
Argument Coherence Engine — thesis mapping + evidence ratio	FULL	—
Argument Coherence Engine — counterargument coverage analysis	FULL	—
Citation & Source Architecture — frequency heatmap + gap detection	FULL	—
Citation & Source Architecture — source recency + primary/secondary balance	CORE	V1 launch
Academic Voice Calibration — register + hedge analysis	FULL	—
Academic Voice Calibration — discipline corpus comparison	DEFER	V1.5
Advisor Dashboard — manuscript list + read-only access	FULL	—
Advisor Dashboard — annotation layer (inline comments)	FULL	—
Advisor Dashboard — progress tracking over time	CORE	V1 launch
Committee Report Generator	FULL	—
LMS integration (LTI 1.3)	DEFER	V1.5
Institutional SSO (SAML 2.0)	DEFER	V1.5

4.2 Academic-Specific Screen Specifications

Screen A1: Academic Onboarding — Role Selection

Element	Behavior
Role selector	After registration, user selects: 'I am a Student / Doctoral Candidate' OR 'I am a Faculty Member / Dissertation Advisor
Discipline selector	Dropdown: Humanities | Social Sciences | STEM | Law | Business | Education | Other. Sets discipline context for all su
Document type selector	Dropdown: Dissertation | Thesis | Journal Article | Book Manuscript | Research Paper | Other Long-Form. Sets analysi
Advisor code field	Optional: student can enter advisor's invitation code to link accounts automatically. If entered, creates advisor_assignm

Screen A2: Argument Coherence Engine

Component	Behavior / Content
Thesis statement card	Extracted thesis displayed prominently. 'Is this your central argument?' Yes / No / Edit. If No: manual entry field. Th
Argument map	Vertical diagram: Thesis at top → branches to chapter-level claims → each claim links to supporting evidence flags
Evidence-to-claim ratio bar chart	Per-chapter bars showing: claim strength (# of claims made) vs. evidence density (# of cited or evidenced stateme
Logical progression map	Horizontal flow diagram showing argument progression chapter-to-chapter. Arrows colored: green (strong connect
Counterargument coverage table	Lists all detected counterarguments with: chapter location | how addressed (Acknowledged/Refuted/Dismissed/Ab
Chapter coherence scores	Table: Chapter | Coherence Score (0-100) | Connection to Prior Chapter | Connection to Next Chapter | Recomme

Screen A3: Citation & Source Architecture

Component	Behavior / Content
Citation frequency heatmap	Grid: rows = authors/works, columns = chapters. Cell color intensity = citation frequency. Hover shows exact coun
Citation gap flags	List of paragraphs/claims with no citation support. Each shows: text excerpt, chapter, AI confidence that citation is
Source recency panel	Bar chart showing citation year distribution. Red zone: publications >10 years old as % of total. Amber: 5-10 years
Format validator	Dropdown to select citation format (APA/MLA/Chicago/AMA). Scans manuscript for format inconsistencies. Lists v

Screen A4: Academic Voice Calibration

Component	Behavior / Content
Register consistency timeline	Chapter-by-chapter line graph: Formality Score (0-10). Dips below 6.0 flagged. Click flagged chapter: opens text e
Hedge analysis panel	Two-column: Over-hedged passages (left, blue) | Under-hedged passages (right, amber). Each entry: text excerpt
Passive voice density bar	Per-chapter passive voice percentage. Red bar if >40%. Click bar: lists all passive constructions in that chapter wi
Voice summary card	Overall academic voice score (0-100). Three sub-scores: Formality, Precision, Authorial Confidence. Short prose s

Screen A5: Advisor Dashboard

Component	Behavior / Content
Student roster panel	Left sidebar: list of all supervised students. Each row: student name, manuscript title, health score badge, last activ
Student manuscript view	Click student: loads their manuscript dashboard in read-only mode. Advisor sees all module results but cannot trig
Annotation tool	Floating toolbar in student manuscript view: 'Add Comment' button. Click opens comment input anchored to the cu
Progress tracking chart	Tab in student view: Health score over time (line chart). X = analysis run dates, Y = overall health score 0-100. Sh
Report generator	Button: 'Generate Committee Report'. Opens modal: select report type (Proposal Defense / Chapter Review / Full
Invite student button	Top of roster: 'Add Student' opens modal with shareable invitation code. Student enters code during onboarding (S

Screen A6: Committee Report Generator

Component	Behavior / Content
Template selector	Four options with preview thumbnails: Proposal Defense | Chapter Review | Full Draft Review | Final Defense Prep
Scope selector	Checkboxes: which modules to include in report. Default: all completed modules.
Advisor notes field	Free-text area: advisor can add narrative comments that appear in the report's Executive Summary section.
Report preview	Live preview pane showing the report structure with placeholder text replaced by actual manuscript data.
Generate & Download	POST /api/v1/manuscripts/{id}/reports with template and options. Returns PDF download. Report header includes

4.3 Academic MVP — Feature Build Status

Feature / Function	Status	Priority	Sprint	Dev Notes
Role selection onboarding flow	Build	P0	S3	Extends base registration; adds discipline + document type
Argument Coherence Engine — thesis extraction	Build	P0	S4	Claude API prompt; thesis pinning UI
Argument Coherence Engine — evidence-to-claim ratio	Build	P0	S4	Per-chapter claim/evidence scoring
Argument Coherence Engine — logical progression map	Build	P0	S5	Graph visualization with gap detection
Counterargument coverage analysis	Build	P1	S5	Detection + strength rating
Citation frequency heatmap	Build	P0	S5	Grid chart; >30% flag logic
Citation gap detection	Build	P0	S5	Claims-without-citation detection pass
Citation format validator (APA/MLA/Chicago/ASA)	Build	P1	S6	Regex + AI hybrid validation
Academic Voice register consistency	Build	P0	S4	Formality scoring per chapter
Hedge analysis (over/under-hedging)	Build	P1	S5	Hedge pattern detection
Passive voice density analysis	Build	P1	S5	Passive detection; rewrite suggestions
Advisor account type + assignment model	Build	P0	S3	advisor_assignments table; role flag
Advisor Dashboard — student roster + read-only	Build	P0	S5	Read-only manuscript view for advisor
Advisor annotation layer (inline comments)	Build	P0	S6	Comments stored with advisor_user_id
Committee Report Generator (PDF)	Build	P1	S6	reportlab template; 4 report types
Student invitation / account linking	Build	P1	S5	Invitation code generation + redemption
Progress tracking chart (score over time)	Build	P2	S7	Longitudinal score charting
LMS integration (LTI 1.3)	Defer	P2	V1.5	Requires institutional partner agreement first
Institutional SSO (SAML 2.0)	Defer	P2	V1.5	Defer until pilot converts to paid contract
Discipline corpus comparison	Defer	P3	V2.0	Requires corpus dataset build

SECTION 5 — ENTERPRISE PUBLISHING MVP

Refinery Enterprise | Acquisition Triage + Multi-Editor Workflow; API-ready for pilot

5.1 MVP Scope Definition

The Enterprise Publishing MVP is designed to support one signed pilot client — a literary agency or small-to-mid-size publisher — within the V1 launch window (Month 7–9). The MVP focuses on the acquisition triage workflow: batch upload, auto-scoring, reader report generation, and multi-editor annotation. Custom rubrics and local hosting are deferred to V1.5.

Feature	MVP Scope	Defer to
Batch upload (up to 50 manuscripts simultaneously)	FULL	—
Auto-scoring — Acquisition Score (0-100)	FULL	—
One-click reader report (PDF)	FULL	—
Priority queue — ranked by Acquisition Score	FULL	—
Multi-editor seats (up to 10 for Standard tier)	FULL	—
Role-based access (Reader, Editor, Acquisitions Director)	FULL	—
Annotation layer — shared editor comments	FULL	—
Decision workflow (Reader → Editor → Director)	FULL	—
Rejection letter drafting	CORE — template-based with AI personalization	Full customization V1.5
Custom scoring rubrics (house voice calibration)	DEFER	V1.5
Genre comparables database	DEFER	V1.5
Market fit scoring add-on	DEFER	V2.0
REST API for external submission ingestion	CORE — webhook receiver only	Full REST API V1.5
Local hosting / air-gapped deployment	DEFER	V1.5
Query letter pre-screening	DEFER	V1.5

5.2 Enterprise-Specific Screen Specifications

Screen E1: Enterprise Org Setup & Seat Management

Component	Behavior
Org profile	Company name, logo upload, primary contact, billing info. Stripe enterprise billing (invoiced, not card-only).
Seat manager	Table: User Name | Email | Role | Status | Last Active | Actions (Edit / Remove). 'Add User' button opens invite modal

Component	Behavior
Role definitions display	Info panel: Reader (can view + annotate; cannot make decisions) | Editor (can view, annotate, move to Director) | Acq
API key panel	Generate/revoke webhook API keys. Shows webhook endpoint URL for submission integration. Docs link.

Screen E2: Submission Intake & Batch Upload

Component	Behavior / Content
Batch upload zone	Multi-file drag-and-drop. Accepts up to 50 files simultaneously. Progress bar per file. Files can be .docx, .txt, .pdf. A
Batch metadata form	Applied to all uploaded manuscripts: Submission round label (e.g., 'Q1 2026 Slush'), Genre filter, Target list (dropd
Individual metadata override	After upload: table shows all queued manuscripts with editable Title, Author Name, Genre, Word Count (auto-dete
Analyze batch button	POST /api/v1/enterprise/batches with manuscript_ids[]. Triggers analysis queue for all. Shows batch progress das
Webhook intake log	Tab: shows all manuscripts received via webhook in last 30 days. Status column shows auto-analysis status.

Screen E3: Acquisition Triage Dashboard

Component	Behavior / Content
Submission queue table	Sortable columns: Acquisition Score | Title | Author | Genre | Word Count | Submitted | Assigned To | Status. Defa
Score visualization	Acquisition Score shown as color-coded badge: 80-100 (green, 'Strong Consider') | 60-79 (blue, 'Consider') | 40-59
Filter bar	Filter by: Genre | Score range | Status (Unreviewed/In Review/Recommended/Passed) | Assigned editor | Date ra
Batch actions	Select multiple rows: Assign to editor | Mark as Pass | Generate reader reports (batch) | Export to CSV.
Analytics strip	Top of page: total submissions | % analyzed | avg score this batch | top genre | time since oldest unreviewed subm
Row click → Manuscript Detail	Click any row opens Manuscript Detail panel (Screen E4) as right-side drawer or full page.

Screen E4: Manuscript Detail — Enterprise View

Component	Behavior / Content
Acquisition Score breakdown	Donut chart showing score composition: Structure (25%) | Voice (20%) | Pacing (20%) | Originality (20%) | Market
Auto-generated synopsis	AI-generated 200-word synopsis of the manuscript. Labeled 'AI Synopsis — verify before use.'
Module summary cards	5 compact cards (same as Indie dashboard but read-only context): Structure | Voice | Pacing | Character | Prose. E
Editor annotation panel	Right column: chronological list of all editor comments with editor name + timestamp. New comment input at botto
Decision workflow bar	Horizontal stepper: Unreviewed → Reader Reviewed → Editor Recommended → Director Decision. Current stage
Reader report button	Generate Reader Report (PDF). See Screen E5.
Rejection letter button	Generate Rejection Letter (see Screen E6). Available after Director marks as Pass.

Screen E5: Reader Report Generator

Component	Behavior / Content
Report template	Matches the house's standard reader report format. MVP ships with one default template; custom templates in V1.
Auto-populated sections	Title & Author | Synopsis (AI-generated, editable) | Strengths (top 3 from module analysis, prose-rendered) | Conc
Editor override fields	All auto-populated sections editable before generating. 'Restore AI Draft' button if editor overwrites.
Generate & Attach	POST /api/v1/manuscripts/{id}/reports/reader. PDF attached to manuscript record. Visible to all editors with access

Screen E6: Rejection Letter Drafting

Component	Behavior / Content
Tone selector	Radio: Standard (professional, brief) | Encouraging (notes strengths before pass) | Detailed (specific craft notes). T
Auto-draft	AI generates personalized letter referencing: manuscript title, author name (from metadata), 2-3 specific strengths
Editor review & edit	Full rich-text editor for the letter. AI draft pre-populated.
Send via / Copy options	'Copy to clipboard' | 'Download as .docx' | (Email integration: post-MVP).
Log action	Sending/copying logs a 'Pass' action in the decision workflow. Updates manuscript status.

5.3 Acquisition Score — Scoring Algorithm

The Acquisition Score is a composite 0–100 score computed from the six Refinery modules. For the MVP, the scoring algorithm uses default weights configurable in a future admin panel:

Component	Default Weight	Source Module	Scoring Logic
Structural Integrity	25%	Manuscript Intelligence Engine	Duplication penalty + chapter coherence score. A manuscript with >20% du
Voice Distinctiveness	20%	Voice Isolation Lab	Average character voice similarity score inverted (high similarity = low score
Pacing Quality	20%	Pacing Architect	Tension curve shape score. Ideal: rising tension with breathing spaces. Flat
Prose Craft	20%	Prose Refinery	Filter word density + show-vs-tell ratio + tic frequency. Inverse scoring: lowe
Narrative Originality	15%	Cross-module AI assessment	AI holistic assessment of premise originality vs. genre tropes. Scale 0-100; v

Note: Custom weight configuration per genre and per house editorial preference is a V1.5 feature (Custom Scoring Rubrics). For MVP, all clients use the default weights shown above.

5.4 Enterprise MVP — Feature Build Status

Feature / Function	Status	Priority	Sprint	Dev Notes
Enterprise account + org model	Build	P0	S4	org_id multi-tenancy; seat model
Role-based access control (4 roles)	Build	P0	S4	Middleware role checks on all enterprise endpoints
Batch upload (up to 50 files)	Build	P0	S5	Queue-based; Celery batch job

Feature / Function	Status	Priority	Sprint	Dev Notes
Acquisition Score algorithm (default weights)	Build	P0	S5	Cross-module composite; configurable weights table
Triage dashboard — sortable queue	Build	P0	S5	Priority sort by Acquisition Score
Manuscript Detail view — enterprise	Build	P0	S5	Score breakdown + editor annotations
Multi-editor annotation layer	Build	P0	S5	Shared comments; @mention notifications
Decision workflow (3-stage stepper)	Build	P0	S6	Role-gated stage advancement
Reader Report generator (PDF)	Build	P0	S6	reportlab; default template; editable
Rejection letter AI drafting	Build	P1	S6	3 tone templates; editor review
Webhook receiver for submission intake	Build	P1	S6	POST /webhooks/submission; API key auth
Batch actions (assign, pass, export CSV)	Build	P1	S6	Multi-select table; batch operations
Analytics strip on triage dashboard	Build	P2	S7	Aggregate stats; basic charts
Custom scoring rubric (house calibration)	Defer	P2	V1.5	Weight config UI + genre profiles
Full REST API (external integrations)	Defer	P2	V1.5	Submittable + QueryManager connectors
Local hosting / Docker deployment	Defer	P1	V1.5	Enterprise security requirement
Query letter pre-screening	Defer	P2	V1.5	Requires separate first-pages + query prompt
Genre comparables database	Defer	P3	V2.0	Requires curated title dataset

SECTION 6 — 12-WEEK SPRINT PLAN

The following sprint plan covers the Alpha MVP build cycle: 12 weeks, 6 two-week sprints. Each sprint has a clear goal, defined deliverables, and acceptance criteria. The first pitchable demo is targeted for end of Sprint 5 (Week 10). Enterprise MVP extends into Sprint 7–8 (Weeks 14–16).

Sprint	Weeks	Goal	Key Deliverables	Exit Criteria
Sprint 1	1–2	Foundation	Repo setup, CI/CD pipeline, FastAPI skeleton, [illegible]	[illegible]
Sprint 2	3–4	Ingestion + Graph	Text extraction pipeline (.docx primary), chapter [illegible]	[illegible]
Sprint 3	5–6	Core Analysis I	Manuscript Intelligence Engine (duplication, cha[illegible]	[illegible]
Sprint 4	7–8	Core Analysis II	Voice Isolation Lab (dialogue extraction, voice fi[illegible]	[illegible]
Sprint 5	9–10	Dashboard + Demo	Manuscript Health Dashboard (all module cards[illegible]	[illegible]
Sprint 6	11–12	Polish + Beta Prep	Export DOCX tracked changes, Committee Rep[illegible]	[illegible]

Note: Sprint 7–8 (Weeks 13–16) covers: Enterprise multi-editor workflow, decision workflow, rejection letter, webhook intake, Pacing Architect scene-level annotations, and Character Arc Workshop basic tracking. These are beta milestone deliverables, not MVP.

SECTION 7 — TESTING & QA REQUIREMENTS

7.1 Unit Testing Requirements

Component	Test Coverage Target	Key Test Cases
Text extraction pipeline	95%	Valid .docx; corrupted file; zero chapters detected; 200K word file; special characters; smart qu
Chapter detection heuristics	90%	Explicit Heading 1 styles; regex chapter markers; ALL CAPS headers; no chapter markers (falll
Character entity extraction	85%	Novel with 20 named characters; single POV manuscript; manuscript with no dialogue; charact
Acquisition Score algorithm	95%	Perfect score inputs; zero score inputs; missing module inputs (partial analysis); all weight com
Authentication & authorization	100%	JWT expiry; refresh token rotation; role permission boundary tests; org_id scoping (no cross-or
File size + type validation	100%	Exactly 5MB; 5.1MB (reject); .exe file (reject); .docx with no text; password-protected .docx (err
Analysis result storage + retrieval	80%	Concurrent analysis runs; result versioning; cache hit vs. cache miss; partial module completion

7.2 Integration Tests

- End-to-end analysis pipeline: upload → extract → graph → analyze → store → retrieve (happy path)
- End-to-end analysis pipeline: error injection at each stage; verify graceful degradation and error state UI
- Multi-user enterprise scenario: 3 editors annotating the same manuscript simultaneously; verify no annotation conflicts or data loss
- Advisor-student account linking: invite code generation, redemption, and manuscript access verification
- Export integrity: exported .docx accepted changes match the accepted items in edit_queue_items table
- Stripe webhook: subscription upgrade/downgrade correctly updates account.tier and feature flags
- Rate limiting: 6 requests/second per user does not cause 500 errors; graceful 429 with retry-after header

7.3 Performance Benchmarks

Metric	Target	Measurement Method
80K-word .docx text extraction + chapter detection	< 15 seconds	Celery task timer; test with 10 different manuscripts
Neo4j character graph build	< 30 seconds	Celery task timer; test manuscript with 20+ named characters
Full 6-module analysis (80K words)	< 3 minutes total	End-to-end timer; median across 5 test runs
Health dashboard load (all results cached)	< 1.5 seconds	Browser performance API; Lighthouse score >85
Pulse graph render (all chapters)	< 500ms	React Profiler; test with 40-chapter manuscript
Enterprise batch of 20 manuscripts	< 60 minutes total	Queue completion timer; run overnight test
Concurrent users (MVP target)	50 simultaneous without degradation	Load test; P95 response time <3s

7.4 Beta Acceptance Criteria

The following criteria must be met before inviting beta users:

- All P0 features for the relevant product line fully functional with no known blocking bugs
- Full analysis pipeline (upload → dashboard) completes successfully for 5 different test manuscripts of varying length and genre
- Zero data leakage between test user accounts (verified by security test)
- Founder has personally completed full end-to-end workflow for each product line using a real manuscript
- Error states all handled gracefully (no unhandled exceptions reaching the user; all errors logged to Sentry)
- Export functionality produces a valid .docx that opens correctly in Microsoft Word and Google Docs
- Mobile responsive check: dashboard fully readable on iPad (tablet) sized viewport

SECTION 8 — MVP EXIT CRITERIA

The MVP is considered pitchable when all criteria in all three columns below are met. 'Pitchable' means: the founder can walk an investor, a prospective institutional client, or a Kickstarter backer through a live demo without hitting a blocking error, incomplete feature, or unexplained gap.

8.1 Indie Author MVP — Pitchable Checklist

Criteria	Status Check	Validated By
Upload a .docx manuscript and receive processing confirmation	Pass / Fail	Founder
Health Dashboard loads with all 5 dimension scores populated	Pass / Fail	Founder
Manuscript Intelligence Engine: duplication detection finds a planted duplicate paragraph	Pass / Fail	Founder
Prose Refinery: filter word counts are accurate vs. manual count in Word	Pass / Fail	QA
Voice Isolation Lab: dialogue extracted correctly for 3 named characters	Pass / Fail	Founder
Pacing Architect: pulse graph renders without error for 30+ chapter manuscript	Pass / Fail	QA
Revision Command Center: accepting an item removes it from pending queue	Pass / Fail	QA
Export: clean .docx downloads and opens in Word without corruption	Pass / Fail	Founder
Free vs. Pro gating: locked modules show upgrade prompt, not error	Pass / Fail	QA
Stripe checkout flow completes successfully in test mode	Pass / Fail	Founder
Full demo run (upload → dashboard → 2 modules → export) in under 5 minutes without crash	Pass / Fail	Founder
2 of 2 indie beta users have completed their first analysis without support intervention	Pass / Fail	Founder

8.2 Academic MVP — Pitchable Checklist

Criteria	Status Check	Validated By
Dissertation upload analyzed correctly with Argument Coherence Engine producing thesis extraction	Pass / Fail	Founder
Evidence-to-claim ratio chart renders accurately vs. manual chapter review	Pass / Fail	QA
Citation frequency heatmap correctly identifies over-cited sources	Pass / Fail	Founder
Academic Voice panel shows formality score per chapter with flagged passages	Pass / Fail	QA
Advisor account successfully created and linked to student account	Pass / Fail	Founder
Advisor can view student manuscript results in read-only mode	Pass / Fail	Founder
Advisor annotation saved and visible in student view	Pass / Fail	QA
Committee Report PDF generated without error for all 4 template types	Pass / Fail	QA
Full demo suitable for Columbus State pilot conversation — no blocking gaps	Pass / Fail	Founder

Criteria	Status Check	Validated By
2 of 2 academic beta users (Ed.D. candidates) have completed first analysis	Pass / Fail	Founder

8.3 Enterprise Publishing MVP — Pitchable Checklist

Criteria	Status Check	Validated By
Batch upload of 10 manuscripts processes without error	Pass / Fail	Founder
Acquisition Score generated for each manuscript with score breakdown visible	Pass / Fail	Founder
Triage queue sorts correctly by Acquisition Score; filter by genre works	Pass / Fail	QA
Two editor accounts annotating same manuscript simultaneously without conflict	Pass / Fail	QA
Decision workflow advances correctly with role gating enforced	Pass / Fail	QA
Reader Report PDF generated with auto-populated synopsis and findings	Pass / Fail	Founder
Rejection letter AI draft generated with manuscript-specific reasons	Pass / Fail	Founder
Webhook endpoint receives test POST and creates manuscript record correctly	Pass / Fail	QA
Full agency/publisher demo run completable in <8 minutes without interruption	Pass / Fail	Founder

8.4 Overall MVP Definition of Done

- All pitchable checklists above: 100% Pass
- No P0 or P1 bugs open in the issue tracker
- Sentry error rate <1% of requests in the 48-hour period prior to demo
- All four beta users have completed onboarding and at least one full analysis
- Founder has run the complete demo script for each product line without errors in a fresh browser session
- The product is accessible at a stable URL (staging or production domain) — not localhost

— END OF FUNCTIONAL DESIGN DOCUMENT —

Refinery | FDD v1.0 | February 2026 | CONFIDENTIAL — FOR DEVELOPMENT USE ONLY