

# Platinum+ AI Design and Implementation SOP

## Blueprint – v3.4.0

### Revision History

Version	Date (UTC-4)	Author	Summary
1.0.0	2025-07-06	Roan	Initial platinum template
2.0.0	2025-07-07	Roan	Foolproof scope, compliance, accessibility, RBAC, SIEM
2.1.0	2025-07-07	Roan	Feature Map, Change Log, phase cadence
3.0.0	2025-07-07	Roan	Documentation law, manifests, anti-pattern guards
3.1.0	2025-07-08	Roan	Drift-correction artefacts, no-deferral doctrine, quarterly audits
3.2.0	2025-07-08	Roan	Historical phase-timeline section
3.2.1	2025-07-08	Roan	Restored Sections 21-23 full content
3.3.0	2025-07-08	Roan	Expanded Sections 1-20 with detail
3.4.0	2025-07-08	Roan	Integrated external benchmark alignment & Design Kit templates; added Completion Prompt Template

## 0. Purpose

Immutable blueprint for MythologIQ / Hearthlink projects. Guarantees crystal-clear scope, auditability, and repeatability. If it is not here or cross-referenced, it is **out of scope**.

## 1. Core Principles

1. **Clarity** – explicit, plain language; zero ambiguity.
2. **User Sovereignty** – data stays local unless owner consents.
3. **Auditability & Traceability** – every action, decision, and data mutation logged, time-stamped, immutable.
4. **Modularity** – components self-contained, versioned, hot-swappable.
5. **Accessibility & Inclusion** – low-stimulation, voice-first, keyboard friendly, colour safe.
6. **Compliance by Design** – GDPR, HIPAA, SOC 2 mapped before code.
7. **Platinum Quality** – zero critical bug, doc gap, or failing test on `main`.
8. **No-Deferral Doctrine** – any discussed feature must be scheduled or implemented before release.

## 2. Roles & Responsibilities (RACI)

Area	Responsible	Accountable	Consulted	Informed
SOP updates	Process Lead	CTO	Security Lead	All devs
Code merge	Module Owner	QA Lead	Security Lead	PM
Compliance validation	Security Lead	CTO	Legal	Dev team
Release sign-off	PM	CTO	QA Lead	Stakeholders

## 3. Documentation Law

### 3.1 Canonical Files

File	Purpose
README.md	Single entry point, module map, quick start, compliance links
/docs/process_refinement.md	Lessons-learned & SOP evolution
/docs/CHANGE_LOG.md	<i>System change log only</i> – time-stamped audit
/docs/FEATURE_MAP.md	Feature inventory & status
/docs/FEATURE_BUILD_PLANS.md	Fully scoped build plans (no deferral)
/docs/UI_COMPONENTS_AUDIT_REPORT.md	Inventory of every UI component
/docs/ASSUMPTIONS_LOG.md	Captured implicit assumptions
/docs/ENVIRONMENT_MANIFEST.md	Supported OS, runtimes, dependencies
/docs/SECURITY_PRIVACY_CHECKLIST.md	Control evidence per phase
/docs/PROMPT_MANIFEST.md	Canonical prompt & agent templates
/docs/DEPRECATIONS.md	Deprecated items & migration paths
/docs/USER_FEEDBACK.md	Synthesised feedback & backlog links
/docs/PLATINUM_BLOCKERS.md	Non-negotiable gating criteria
/docs/ DRIFT_CORRECTION_CASE_STUDY.md	Immutable retrospective

### 3.2 Structural Rules

- Change Log and Process Refinement may **never** overlap content.

- Every canonical file must be referenced from README.
- Doc-lint CI rejects broken links, missing cross-refs, undefined glossary terms, and em dashes.

### 3.3 Cross-Reference Matrix

Each PR must reference at least one Feature Map ID **and** one Change-Log entry. Variance report auto-fails on mismatch.

## 4. Project Planning Standards

1. **Phase kick-off package** – Feature Map cross-check, variance summary, sprint goals, signed test plan.
2. **Branch discipline** – one branch per feature/fix; protected branches require green CI, two reviewers, doc/test coverage gate.
3. **Prompt template rule** – every AI prompt/ticket must cite Feature-Map IDs, doc sections, acceptance tests.
4. **Scope control** – anything outside scope logged in `FEATURE_WISHLIST.md` **and** `ASSUMPTIONS_LOG.md`.
5. **Assumptions review** – kickoff updates assumptions; close verifies none remain implicit.
6. **Dependency review** – ENVIRONMENT\_MANIFEST.md updated every sprint.

## 5. Implementation Standards

1. **Doc-driven development** – no code without approved doc (PRD/blueprint) in `/docs/`.
2. **Test driven** – unit first; negative/edge cases mandatory. Coverage targets: unit 95 %, integration 90 %, e2e 85 %.
3. **Security first** – static + dynamic scans each build; SECURITY\_PRIVACY\_CHECKLIST.md updated.
4. **Config-doc sync** – linter ensures config keys appear in docs and vice versa.
5. **Prompt stability** – PROMPT\_MANIFEST version bumps required on template change; diff validator enforces.
6. **Deprecation path** – code removal requires entry in DEPRECATIONS.md and migration notes.

## 6. Cadence & Rituals

Cadence	Activity	Required Artifacts
Daily	Push, lint, green CI	fresh commits
Weekly	Status sync	updated Feature Map & Change Log
Sprint end	Phase checklist, variance, retro	<code>PHASE_&lt;N&gt;_FEATURE_CHECKLIST.md</code> , retro notes

Cadence	Activity	Required Artifacts
Quarterly	<b>Full audit</b> – Feature Map, UI Audit, Blockers	AUDIT_<YYYY-Q>.md
Release	Tag, full QA, sign-off	PHASE_<N>_RELEASE_SUMMARY.md, Change Log update

No phase closes unless features accounted for, docs current, tests pass/triaged, audit complete.

## 7. Automated Enforcement

- **Git hooks** – lint, test, doc-lint, config-doc sync.
- **CI pipeline** – build → static analysis → tests → doc coverage → container scan.
- **Validators** – feature-map validator, assumption checker, prompt-manifest diff, audit-trail linter.
- **Runtime monitors** – branch discipline, doc compliance. Violations block merge and raise Sentry alert.

## 8. Feature Map & Inventory SOP (Phase 13)

### 8.1 Feature Map Spec

Fields: id, name, type (Primary/Secondary/Deferred/Wishlist), status, source\_refs, owner, notes, last\_updated.

### 8.2 Checklists

- Generate PHASE\_<N>\_FEATURE\_CHECKLIST.md at kickoff **and** close.
- PR template requires one or more IDs.
- Retroactive audit when gaps found; create entries immediately.

### 8.3 Deprecations & Retirements

Deprecated items move to DEPRECATIONS.md with date, reason, migration path; Feature Map status set to retired.

## 9. Security & Privacy Checklist Spec

Tracks encryption at rest/in transit, access control, logging, PII handling, breach response. Evidence attached each phase.

## 10. Environment Manifest Spec

Lists OS versions, runtime versions, external services, library versions, licences. Updated every sprint; CI fails on unsupported drift.

---

## 11. Prompt & Agent Manifest Spec

Stores template text, version, description, owner, last review date. Template change triggers approval workflow and tests.

---

## 12. User-Feedback Loop

After each beta/user test, synthesise findings into USER\_FEEDBACK.md and create backlog items. Phase cannot close without reviewing new feedback.

---

## 13. Deferral & Scope Management

- `FEATURE_WISHLIST.md` stores non-core ideas scored (value 1-5, complexity 1-5).
  - **No-deferral doctrine** – core features cannot be postponed; wishlist items must receive build plan before release if promoted.
  - Change-control form required for any scope adjustment.
- 

## 14. Review & Approval Loop

- End-of-sprint demo → retro → decision-log update.
  - Urgent architecture change → ADR template ( `/docs/adr/` ) with 24-hour review window and quorum of two leads.
  - Decisions logged and linked to Change Log.
- 

## 15. Continuous Improvement

- `scripts/variance_report.py` compares SOP coverage vs repo weekly.
  - Lessons learned appended to `process_refinement.md` via PR.
  - Quarterly kaizen workshop – action items tracked as Jira epics.
-

## 16. Synapse Connection Integration SOP

1. PRD in `/docs/blueprints/synapse_<name>.md` with risk score, benchmark plan.
  2. Feature branch, CI gating, SIEM rules, UI wizard steps.
  3. QA matrix: permission toggles, sandbox, rollback.
  4. Compliance re-evaluation if data leaves device.
- 

## 17. MCP Agent Resource Policy SOP

- Policy file per agent `<agent>.mcp.json` – allow list of resources/verbs.
  - Loaded at agent start; violation → Sentry incident.
  - Policy diff PR; reviewer sign-off mandatory.
- 

## 18. Installation UX & First-Run Experience

- Electron installer, code-signed. Steps: system check → animated persona reveal → mic/speaker calibration → consent screen → personas carousel.
  - Rollback on failure; installer transaction log; self-delete on cancel.
  - Low-stimulation visuals, screen-reader friendly.
- 

## 19. Disaster-Recovery & Performance Requirements

### 19.1 Backup SOP

- Snapshot Vault & config every 15 minutes; retention hourly 24h, daily 14d, weekly 12w, monthly 12m.
- Restore wizard in Sentry UI; owner role + passphrase.

### 19.2 Performance & Sustainability

- Idle RAM < 500 MB on 32 GB rig; max CPU spike 70 % < 2 s; IPC latency < 30 ms.
  - Carbon metric (watts drawn) logged to SIEM.
- 

## 20. Historical Phase Timeline & Numbering Rationale

Phase	Outcome	Key Events
1	Foundation	Modular codebase planned, <code>process_refinement.md</code> created.
2	Early Success	Core modules delivered; discipline forming.

---

Phase	Outcome	Key Events
3	Traceability Surge	Branch/audit rules enforced retroactively.
4–6	Rigor	95 %+ test coverage; SOP codification.
7–9	Drift Detected	>50 % features silently deferred; recognition triggered corrective action.
13	Feature Map Birth	Intentional skip to highlight drift; <code>FEATURE_MAP.md</code> created; no-deferral doctrine conceived.
<b>Return to 10–15</b>		

## 21. External Benchmark Alignment

This SOP now maps directly to industry frameworks. A full cross-walk lives in `/docs/BENCHMARK_CROSSWALK.md` (created 2025-07-08). Highlights:

External Standard	SOP Coverage	Gap / Action
<b>NIST AI RMF 1.0</b>	Governance, Map, Measure, Manage covered via Sections 1–23.	Add AI Impact Assessment form (AIIA) template under <code>/docs/templates/</code> and make it mandatory in Phase kick-off.
<b>ISO/IEC 42001:2023</b>	Lifecycle controls reflected in Security/Privacy Checklist and Continuous Improvement.	Embed clause mapping table in <code>SECURITY_PRIVACY_CHECKLIST.md</code> ; require quarterly management review minutes.
<b>OWASP SAMM 2.1</b>	Secure SDLC enforced by Git hooks, CI pipeline, Platinum Blockers.	Integrate threat-model review (STRIDE) and SBOM scan stage; target maturity Level 2 by Q4-2025.
<b>IEEE 7001-2021</b>	Transparency via Prompt Manifest, Explainability tooling, logs.	Require model cards in <code>/docs/model_cards/</code> , exposed to end-users.
<b>2025 MLOps Roadmaps</b>	CI/CD with Continuous Training pipeline planned.	Add dedicated „Model Ops“ stage to pipeline; configure drift monitor alerts into SIEM.

*All identified actions have Feature Map IDs and Build Plan entries.*

## 22. Design-Kit Template Standard


To replicate this SOP in future projects, clone `/design-kit/` which contains ready-made Markdown and JSON templates:


Template File	Purpose
<code>TEMPLATE_README.md</code>	Pre-filled authoritative README scaffold
<code>TEMPLATE_SOP.md</code>	Empty SOP skeleton referencing Benchmark Crosswalk
<code>TEMPLATE_FEATURE_MAP.md</code>	Blank feature inventory table
<code>TEMPLATE_BUILD_PLAN.md</code>	Per-feature plan incl. timeline, tests
<code>TEMPLATE_UI_AUDIT.md</code>	UI components inventory sheet
<code>TEMPLATE_PHASE_CHECKLIST.md</code>	Kick-off/close checklist
<code>TEMPLATE_AIIA.md</code>	AI Impact Assessment form (ISO 42001)
<code>TEMPLATE_ADR.md</code>	Architecture Decision Record
<code>TEMPLATE_BENCHMARK_CROSSWALK.md</code>	Matrix mapping SOP ↔ standards
<code>TEMPLATE_PROMPT.md</code>	Completion Prompt & Enforcement Checklist (see Sect 26)


**Automation:** `scripts/kit_init.py <project-name>` copies and renames templates, sets `project_id` placeholders, and registers them in `CHANGE_LOG`.

## 23. Completion Prompt & Enforcement Template

Embed the following prompt in `TEMPLATE_PROMPT.md` and surface it in developer docs.

```
#  COMPLETION PROMPT TEMPLATE – SOP ALIGNED

##  PHASE [#]: [PHASE NAME]

###  Task: [TASK TITLE]

**Prompt:**
"Implement/Resolve [brief system-level action]. Ensure strict compliance with
modularity, auditability, and test coverage."

**Reference Docs:**
- `FEATURE_MAP.md` ([Feature ID])
- `appendix_*.md` (as needed)
- `FEATURE_BUILD_PLANS.md`
```



```

- `appendix_h_developer_qa_platinum_checklists.md`

**Actions:**
- Implement feature/module/fix as described
- Build test suite (95%+ coverage unless otherwise noted)
- Update `FEATURE_MAP.md` and `FEATURE_BUILD_PLANS.md`
- Log implementation in `CHANGE_LOG.md`
- If deviations are required, add entry to `variance_reports/[feature-
name]_variance.md`

---

## 🟡 SOP ENFORCEMENT CHECKLIST
* Section 26: No deferrals – build every planned or referenced feature
* Section 27: Every feature must include:
    ✓ Branch
    ✓ Commit
    ✓ Test
    ✓ Documentation
    ✓ Cross-reference in `FEATURE_MAP.md`
* All process evolution → `process_refinement.md`
* All implementation changes → `CHANGE_LOG.md`
* All deviations → `variance_reports/`
* No merges to `main` without audit trail and owner sign-off
---

## 🛑 FINAL STATUS LOGGING
* Update `FEATURE_MAP.md` – set feature status
* Update `FEATURE_BUILD_PLANS.md` – confirm implementation notes
* Update `CHANGE_LOG.md` – include exact commit summary
* Update `variance_reports/` – if applicable
* Update `process_refinement.md` – if SOP evolution required

```

Developers copy this block, fill placeholders, and attach to every PR description. Git-hook parser verifies checklist completion before allowing merge.

## 24. Glossary Enforcement

To prevent jargon drift and ensure shared understanding across contributors, auditors, and future maintainers:

1. **Single Source of Truth** – All domain-specific terms must be defined in `appendix_g_system_glossary_taxonomy.md` (Appendix G).
2. **Automated CI Check** – The doc-lint task parses every capitalised token; the build fails if a term is absent from the glossary or used inconsistently.

3. **Version Tagging** – Each glossary entry contains the SOP version in which the term was introduced plus the date of latest revision.
4. **Alphabetical Order & Style** – Glossary entries are ordered A→Z, include a concise definition ( $\leq 50$  words), and may reference RFCs or standards for precision.
5. **Ownership & Review Cadence** – The Documentation Lead curates the glossary. Contributors propose additions via PR with change-log entry. Glossary review occurs at each quarterly audit.

---

## 25. Usage Guidance

The following guidelines ensure the SOP and its companion documents remain authoritative and developer-friendly:

1. **Initial Cloning** – Copy this file into any new repository as `/docs/SOP.md` before the first commit.
2. **Section Immutability** – Headings must never be removed. If a section is not applicable to a given project, append **N/A** beneath the heading but retain the section for consistency.
3. **Validation Workflow** – On every commit, run `scripts/sop_validator.py` which checks:
  4. Canonical files exist and are referenced.
  5. Glossary, Feature Map, Change Log cross-links are present.
  6. No em dash, broken link, or undefined term. CI blocks the merge on any validator failure.
7. **Contributor Onboarding** – New team members must read `DRIFT_CORRECTION_CASE_STUDY.md` and this SOP in full. The onboarding checklist includes a short quiz to confirm understanding.
8. **Change Control for the SOP** – Any modification to this document:
  9. Requires a dedicated PR labelled `SOP-Update`.
  10. Must reference the related discussion in `process_refinement.md`.
  11. Must update the Revision History table with a succinct summary.
  12. Must include a Change Log entry citing affected sections.

---

## 26. Appendix Map

Appendix	File	Purpose
A	<code>appendix_a_combined_open_items.md</code>	Open items tracker
B	<code>appendix_b_integration_blueprints.md</code>	Integration blueprints
C	<code>appendix_c_ui_blueprints.md</code>	UI blueprints
D	<code>appendix_d_technical_product_requirements_document_technical_prd.md</code>	Technical PRD
E	<code>appendix_e_model_context_protocol_mcp_full_specification.md</code>	MCP full specification

Appendix	File	Purpose
F	<code>appendix_f_end_to_end_data_flow_examples.md</code>	Data-flow examples
G	<code>appendix_g_system_glossary_taxonomy.md</code>	System glossary & taxonomy
H	<code>appendix_h_developer_qa_platinum_checklists.md</code>	Developer & QA checklists
I	<code>/design-kit/TEMPLATE_PROMPT.md</code>	Standard completion prompt template

#### Usage Guidance

*(renumbered from Section 22; content unchanged)*

## 23. Appendix Map

Added **I – Templates & Prompt Examples**.

Appendix	File	Purpose
A	<code>appendix_a_combined_open_items.md</code>	Open items tracker
...	...	...
I	<code>/design-kit/TEMPLATE_PROMPT.md</code>	Standard completion prompt

**End of SOP v3.4.0**