

Problem Statement

During critical security incidents, Abnormal's customers experience delayed and inconsistent communications—undermining trust when clarity matters most. Today's manual process forces Incident Commanders(engineer/ops lead), Engineering, Support, and Legal into slow, error-prone drafting and multi-round reviews, lengthening MTTA/MTTR and distracting engineers from remediation.

- **Manual drafting and review** introduce delays, risks of omission or oversharing, and cognitive overhead in tracking severity-based update cadences.
- **Inconsistent author tone and structure**, coupled with no centralized template library, results in brand drift and confusing customer updates.
- **Sequential approval cycles** and context-switching across PagerDuty, Slack, docs, and Statuspage add additional latency per update and risk version drift.
- **Engineers are pulled away** from urgent remediation to handle communication tasks, slowing resolution and increasing MTTR.

Why Now?: With the rise of complex, unpredictable threats—including those powered by agentic AI—incident frequency and urgency are increasing. Customers now expect near-instant visibility, but current manual processes can't scale to meet that demand.

Goals & Success Metrics

 **Time to First Customer Update:** Target ~50% faster than current baselines


✓ **Why it matters:** A rapid first notice builds customer confidence and reduces inbound support volume during high-severity incidents.

 **Review Iterations:** Target 1 review round across stakeholders

✓ **Why it matters:** Fewer back-and-forths lead to faster turnarounds, reduced stakeholder friction, and more time spent on actual remediation.

 **P0 Incident Adoption:** From 0% → 100% within 3 months

✓ **Why it matters:** Full adoption for the most urgent incidents ensures we're delivering efficiency and consistency when stakes are highest—and sets the foundation for broader rollout.

 **Tone Consistency:** ≥ 95% on automated style-check audits

✓ **Why it matters:** A unified, on-brand voice reassures customers, reduces ambiguity, and minimizes the risk of over- or under-communicating impact.

Key Personas & User Journeys

Incident Commander (Engineer/Ops Lead)

Role: Coordinates triage, classification, and team execution

Emotional Nuance: Under intense pressure to respond quickly and accurately

Needs:

- Speed and clarity in generating comms
- Ability to manage approval flow across stakeholders
- Full visibility into status of drafts and publish timing

Workflow Interaction:

- Triggers LLM draft generation
- Oversees and approves final customer-facing update
- May initiate follow-up updates via Slack or dashboard UI

Customer Support Lead

Role: Ensures message clarity, empathy, and customer-appropriate language

Emotional Nuance: Eager to reassure customers but wary of miscommunication

Needs:

- Confidence in tone and structure of updates
- Fast way to review or suggest changes

Workflow Interaction:

- Reviews/edit AI drafts for clarity
- Can re-prompt or request simplified language
- Co-appraiser on message tone for publish-ready drafts

Legal / Compliance Reviewer

Role: Ensures compliance, regulatory alignment, and safe public messaging

Emotional Nuance: Cautious about legal risks and information leaks

Needs:

- Redaction of sensitive/internal data
- Clear version control and approval history

Workflow Interaction:

- Receives AI-highlighted “risky” terms
- Signs off on final version or requests revision
- May trigger blocked publish if compliance concerns are found

Resolving Engineer (Technical Responder)

Role: Diagnoses root cause, deploys fixes, confirms recovery

Emotional Nuance: Focused on resolving issues, frustrated by interruptions

Needs: Minimal distractions during incident resolution

Workflow Interaction:

- Provides structured status updates via Slack/Jira
- May confirm technical accuracy in draft, but not a primary reviewer
- Beneficiary of a system that reduces Slack pings and “can you review this message?” interruptions

CISO / Executive Reviewer (Secondary Persona)

Role: Provides high-level oversight and sign-off on critical incident communications within Abnormal Security leadership

Emotional Nuance: Concerned with organizational reputation and risk management

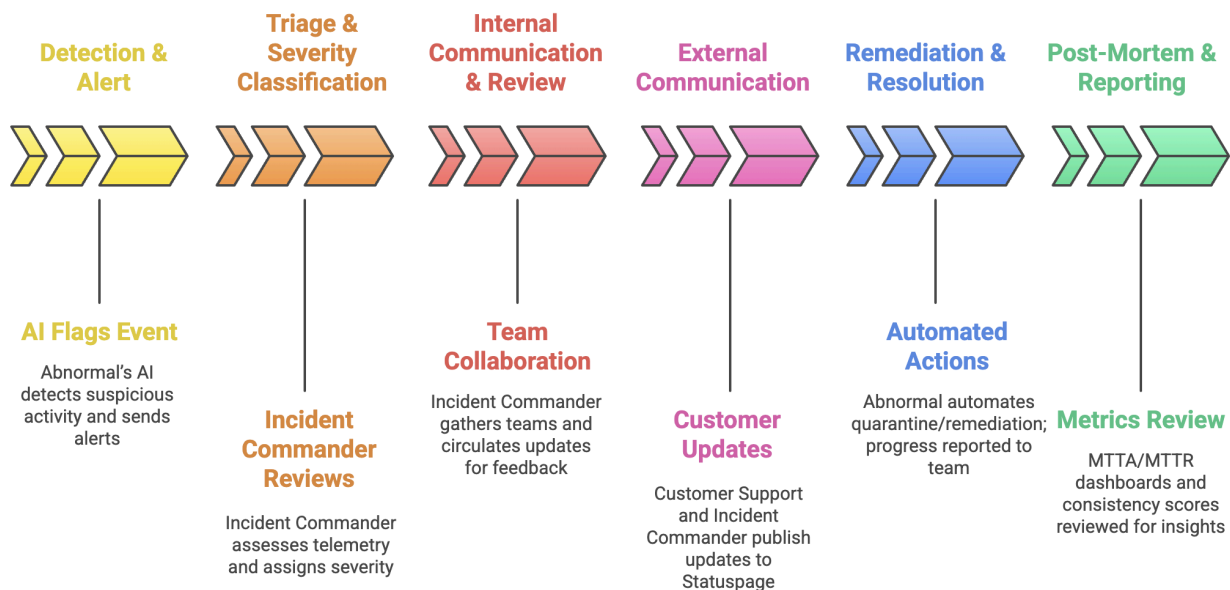
Needs:

- Real-time visibility into incident severity and messaging
- Ability to approve or escalate messaging for high-impact incidents

Workflow Interaction:

- Views drafts in read-only mode
- Provides final sign-off trigger for P0 incidents
- Escalates to broader leadership if necessary

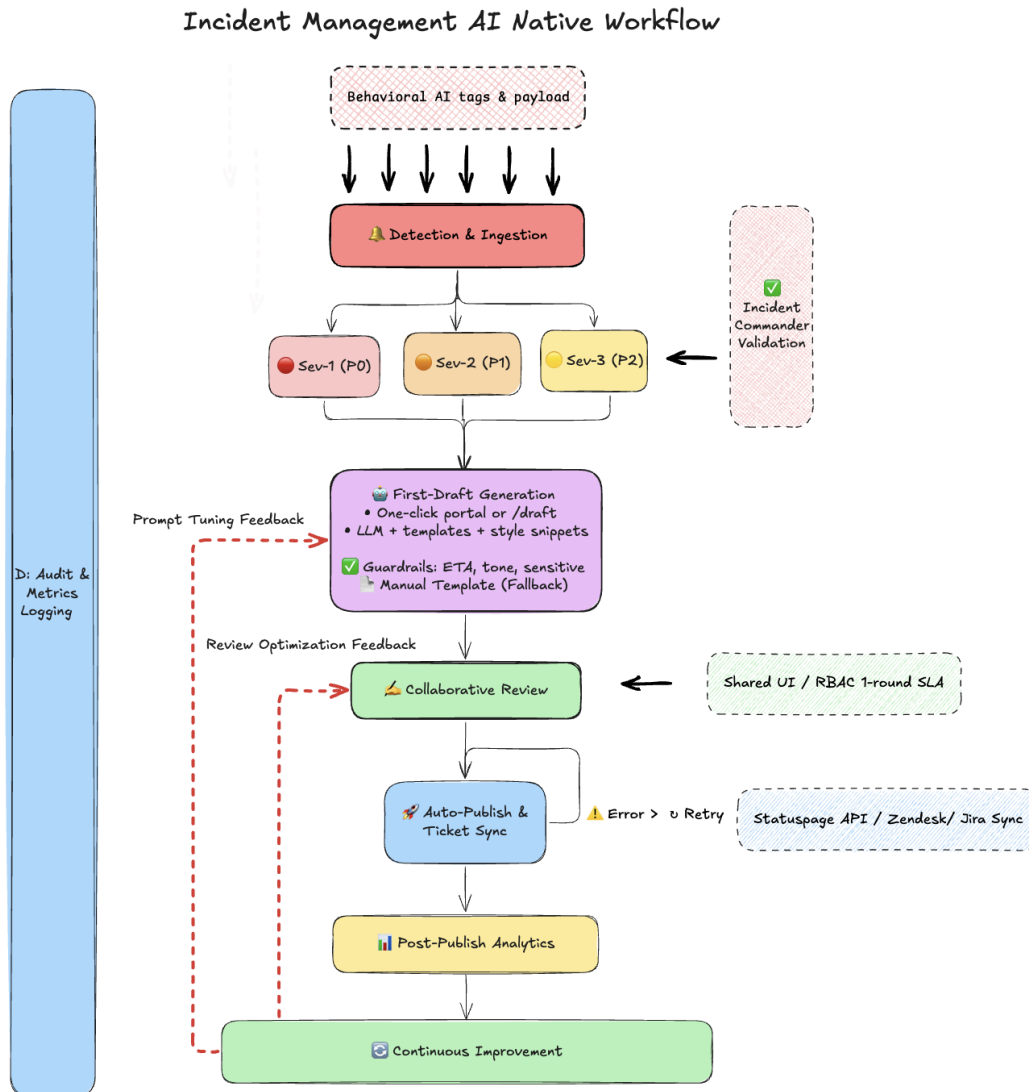
Incident Management Lifecycle Stages



Current Incident Management Lifecycle - Stages

Stage	Abnormal Context	Pain Points
Detection & Alert	Abnormal's behavioral AI flags a suspicious event (e.g., BEC attempt, credential anomaly). Alert is sent via PagerDuty/Slack/Email to on-call engineers and support.	<ul style="list-style-type: none"> No auto-prioritization → all alerts look equally urgent Lack of contextual summary → engineers sift raw logs
Triage & Severity Classification	Owner: Incident Commander (engineer/ops lead) reviews telemetry & context to assign severity (P0 - Critical, P1, P2 etc)	<ul style="list-style-type: none"> Inconsistent severity thresholds under pressure Under- or over-classification drives wrong update cadence
Internal (Stakeholder) Communication + Review & Approval	Incident Commander gathers engineering, support, and Customer Success teams in Slack/war-room to share high-level context. Draft message is circulated to Engineering Lead, Legal/Compliance, and Customer Success via email threads or shared docs for feedback.	<ul style="list-style-type: none"> Ad-hoc updates—some stakeholders out of the loop Manual duplication of notes across tools (Slack → email → docs) Sequential review bottlenecks Version confusion and late approvals slow publishing
External (Customer) Communication	Customer Support and Incident Commander draft, approve, and publish updates to Statuspage.io; notifications sent via email/SMS/webhooks.	<ul style="list-style-type: none"> Manual cut-and-paste errors into Statuspage Tone and format vary by author (too technical vs. too vague)
Remediation & Resolution	Abnormal automates quarantine/remediation actions; progress is reported back to the incident team.	<ul style="list-style-type: none"> Engineers pulled into verifying comms Updates risk falling out of sync with actual remediation progress
Post-Mortem & Reporting	MTTA/MTTR dashboards and a “consistency score” track time to first customer update, resolution time, and tone consistency.	<ul style="list-style-type: none"> Insights reviewed after the fact, not fed back into playbooks Lack of real-time metrics to adjust ongoing incident workflows

Proposed AI-Native Workflow



Step 1: Detection & Ingestion

Today (Manual): Alerts fire in PagerDuty/Slack and engineers must parse raw logs to figure out severity, scope, and timeline (often cutting & pasting across tools).

AI-Native:

- **Automated Tagging & Payload:** Abnormal's AI assigns a preliminary severity (P0–P3), impacted components, confidence score, and timestamp.
- **Commander Validation:** Incident Commander confirms or adjusts that tag in-portal or in the PagerDuty alert.
- **API Context:** Finalized payload flows via Incident-API webhook (P0 real-time) or polling (P1–P3) into the workflow within 30 s–15 min depending on severity.

Step 2: First-Draft Generation

Today (Manual): Support Lead hand-writes a 1–2-paragraph update by copying facts from internal notes and wrestling with tone/accuracy (5–10 min).

AI-Native:

- **One-Click Draft (< 5 s):** Commander hits “Generate Draft” in the portal or `/draft` in Slack.
- **Unified Output:** LLM uses the confirmed payload (severity, systems, ETA) plus style-guide snippets to emit:
 - **Internal summary** for engineering
 - **Customer-facing update** formatted for Statuspage
- **Guardrails & Fallback:** Auto-checks for missing ETA, sensitive terms, or tone drift; falls back to a minimal manual template if checks fail—no copy-paste or version splits.

Step 3: Collaborative Review & Editing

Today (Manual): Draft circulates via email/Slack/Confluence; Support, Legal, Commander each edit in separate threads, creating 2–3 rounds of back-and-forth (30–60 min).

AI-Native:

- **Shared Workspace:** Single web UI (or Slack thread) where all reviewers see and edit the same draft.
- **AI Suggestions:** In-context flags for missing fields, off-brand language, or compliance risks; one-click “apply” fixes.
- **Fast One-Round SLA:** Automated nudges at 50% of time-box (e.g. 5 min for P0) ensure all designated approvers finish in one pass.
- **RBAC Control:** Only authorized roles can approve or block the publish.
- After AI suggestions, show a “Rate Draft” control. Collect reviewer scores & tags alongside their edits.

Step 4: Auto-Publish & Ticket Sync

Today (Manual): Final text is copy-pasted into Statuspage and separate ticketing systems, risking errors and extra effort.

AI-Native:

- **One-Click Publish:** Approved update flows via Statuspage API to the right component(s) and subscriber channels.
- **Ticket Sync:** The same message pushes into Zendesk/Jira for audit and customer-service context—fully automated, zero extra work.
- **Retry Logic:** Automatic retry up to 3× with exponential backoff on API errors; fallback shows a manual “Retry Publish” button if automated attempts fail.

Step 5: Post-Publish Analytics & Executive Visibility

Today (Manual): MTTA/MTTR and review metrics live in spreadsheets; leadership gets ad hoc email updates.

AI-Native:

- **Real-Time Dashboard:** Logs every draft, review, and publish; surfaces Time to First Update, Review Iterations, Tone Score, and P0 Adoption.
- **Executive Summaries:** CISO and execs receive automated, high-level incident summaries immediately after any P0 event.

Step 6: Continuous Improvement & Feedback

Today (Manual): PMs occasionally tweak templates after post-mortems; drift isn't proactively tracked.

AI-Native:

- **Analytics-Driven Refinement:** Automated drift detection flags prompt or style deviations.
- **Auto-Tuning:** Template and prompt library updates push live based on real-world performance data—closing the loop without manual intervention.

Augmented RHLF Loop:

1. **Ingest Review Data:** Pull edit diffs and ratings from the review workspace.
2. **Ingest Customer Signals:** Pull simple ✓/✗ or emoji reactions from status-page subscribers or support tickets.
3. **Retrain** (or re-prompt): Weekly, use these labeled examples to adjust prompt templates or fine-tune a small reward-model layer.
4. **Deploy Updates:** Automatically roll out improved prompt snippets and style guards to the Draft Generation service.

Functional Requirements

Requirement ID	Abnormal Context	Pain Points
FR-1	Incident Ingestion	Connect to Incident API / PagerDuty to pull severity, components, timestamp payloads.
FR-2	Draft Generation Endpoint	POST /draft returns LLM-generated internal & public messages in < 5 s with guardrail flags.
FR-3	Review Workspace UI	Shared web interface (and Slack thread) for inline comments, AI “apply suggestion” buttons, and 1-round SLA timer.
FR-4	RBAC & Approval Controls	Enforce roles: only Incident Commander, Support, and Legal can approve or block publish.
FR-5	Publish Integration	One-click publish to Statuspage via REST, plus automatic ticket sync to Zendesk/Jira.
FR-6	Audit Log Service	Record every draft, review action, approval decision, and publish event in an append-only store.
FR-7	Real-Time Metrics Dashboard	Surface Time-to-First-Update, Review Iterations, Tone Score, and P0 Adoption—driven by the audit log.

Non - Functional Requirements

Requirement ID	Abnormal Context	Pain Points
NFR-1	Performance (Draft Latency)	95% of /draft calls complete in < 5 s.
NFR-2	Availability & Reliability	99.9% uptime for core services; automated retry logic for publish failures.
NFR-3	Security & Compliance	Data encrypted in transit & at rest; audit logs immutable; GDPR-friendly retention.
NFR-4	Scale & Throughput	Support concurrent handling of 100 simultaneous incidents without degradation.
NFR-5	Monitoring & Alerting	Auto-alert on SLA misses (e.g. first update > 15 min), draft failures, or API errors.

AI System Design & Guardrails

Overview

The AI architecture combines LLM prompt templating, automated validation, and continuous feedback loops to ensure that every incident communication is fast, accurate, and on-brand.

A. Prompt Shaping & Template Management

- **Severity-Specific Templates:** Store and version-control a library of prompts tuned for P0–P3 incidents, each including:
 - Incident context boilerplate (e.g. “We are actively investigating...”)
 - Customer impact framing and next-step placeholders
 - Closing reassurance language
- **Style-Guide Injection:** Embed Abnormal’s tone and terminology rules as system-level instructions or few-shot examples, ensuring consistent brand voice.
- **Dynamic Slot Filling:** Systematically insert structured fields (severity, impacted components, ETA) into prompts to minimize hallucination.
- **Versioning & Auditability:** All prompt templates are stored in a Git-based repo with change logs, enabling rollbacks and audit trails.

B. Guardrails & Validation Checks

- **Field Completeness:** Verify presence of ETA, impacted components, and severity in each draft.
- **Sensitive Content Filtering:** Block or redact internal hostnames, PII, legal identifiers, and raw stack traces.
- **Tone Consistency Scoring:** Use an automated style-scoring service to enforce ≥95% compliance with brand guidelines.
- **Fallback Strategy:** On any validation failure or LLM timeout, automatically provide a minimal manual template (fields-only) to keep the process moving—no blank screens or unvetted drafts.

C. Human-in-the-Loop Feedback (RHLP)

- **Reviewer Ratings:** Capture 👍/👎 or 1–5★ feedback in the review UI; tag examples as “good” or “needs work.”
- **Customer Signals:** Ingest simple customer reactions (emoji feedback, support-ticket flags) post-publish to gauge clarity and impact.
- **Drift Detection & Alerts:** Monitor aggregate style scores and feedback trends; trigger alerts if metrics fall below SLAs (e.g., style score <90% for three consecutive drafts).
- **Continuous Prompt Tuning:** Weekly, the ML team curates top-rated vs. flagged examples to refine prompt templates and guardrail patterns, then deploys updates automatically into the Draft Generation service.

Security & Compliance Note: All prompts and drafts are processed in an encrypted environment. Sensitive payload data is redacted in logs, and audit logs are immutable to meet enterprise compliance standards.

Model Selection & Evaluation

Model Choice

Primary: GPT-4o via Azure OpenAI (highest context window, enterprise SLA, SOC 2 compliance).

Fallback: Internal fine-tuned Mixtral-8x7B hosted in VPC for sensitive data regions (EU residency).

Promotion gate: Model must achieve $\geq 95\%$ Tone-Consistency and $\leq 1\%$ jailbreak in staging before prod cut-over.

Evaluation Framework

Layer	Metric	Target	Dataset	Judge / Tool
LLM-as-Judge (automated)	Tone Consistency	$\geq 95\%$	200 “gold” incident updates	GPT-4o critique prompt + cosine-sim Δ
	Content Accuracy (facts)	$\geq 98\%$	50 synthetic incidents w/ ground-truth JSON	LLM-based extraction vs. ground truth
	Policy / Jailbreak	$\leq 1\%$ success	100 adversarial prompts	Anthropic red-team + automated policy judge
Human-in-Loop (shadow)	Edit Distance vs. Final	$\leq 10\%$ words changed	Live P0–P2 incidents (2 wks)	Review UI diff logger
	Draft Latency	$< 5\text{ s P95}$	Live	API timer

Risks & Mitigation

Risk	Impact	Mitigation
AI Hallucinations & Inaccurate Drafts	Misleading or incorrect customer updates; trust erosion	<ul style="list-style-type: none">• Strict guardrails (field completeness, sensitive-term filters)• Human-in-loop review for all P0/P1 drafts• Fallback manual template when checks fail
Prompt Drift & Style Inconsistency	Tone fluctuations over time; brand misalignment	<ul style="list-style-type: none">• Automated style-consistency scoring with alerts (threshold < 90%)• Weekly prompt-template tuning using reviewer feedback
Publish Failures / Integration Errors	Updates never reach customers; SLA breaches	<ul style="list-style-type: none">• Automated retry logic (×3 exponential backoff)• Manual “Retry Publish” button• Real-time publish-error alerts
SLA Misses (Late Updates)	Customer dissatisfaction; churn risk	<ul style="list-style-type: none">• SLA-driven nudges in review UI• Dashboard monitors “time to first update” with alerting on any breach
Low Adoption / Workflow Bypass	Teams revert to manual process; investment wasted	<ul style="list-style-type: none">• Mandatory enforcement for all P0 incidents via RBAC• Training & onboarding sessions• Success metrics tied to performance reviews
Data Privacy & Compliance Exposures	Leaks of PII or internal data; regulatory violations	<ul style="list-style-type: none">• Automatic redaction of PII/internal identifiers• Immutable, encrypted audit logs• Role-based access control (RBAC)
Scalability & Performance Bottlenecks	Slow draft generation or UI timeouts under load	<ul style="list-style-type: none">• Performance SLOs (95% drafts < 5 s)• Horizontal scaling of LLM service• Load-testing in staging before rollout
Prompt-Injection / Jailbreak	Malicious or crafted input could force the LLM to reveal internal data, ignore guardrails, or output off-brand / non-compliant text.	<ul style="list-style-type: none">• Input Sanitization – strip Markdown/HTML, escape special tokens, and refuse system-level override strings.• Jailbreak Test Suite – nightly automated prompts (e.g., “Ignore all previous instructions...”) must score < 1 % jailbreak success before deployment.• Runtime Detection – pattern-match for jailbreak markers (e.g., “as an AI language model”) and auto-fallback to manual template if triggered.