VERTEX, INC. | PRODUCT MANAGEMENT

Product Requirements Document

Apex Smart Alerts

Proactive anomaly detection for the Apex Analytics platform

Document Control

Document owner	Maya Chen, Senior Product Manager, Apex Platform
Status	Draft for Review
Version	1.3
Last updated	March 9, 2026
Target release	Apex Q3 2026 (v4.2)
Contributors	Daniel Ruiz (Engineering), Priya Nair (Design), Omar Haddad (Data Science), Lena Fischer (Customer Success)
Reviewers	Sofia Alvarez (VP Product), Growth & Enterprise Eng leads

At a glance

Apex customers miss revenue-impacting metric shifts because dashboards require active monitoring. Smart Alerts pushes anomaly detection to users automatically, closing the gap between *data available* and *data acted on*.

Related documents:
Apex Alerting Technical RFC (ENG-2026-014)
Q3 Retention Roadmap (Growth-PRD-009)
Enterprise Notification Preferences Spec

This document is confidential and intended for internal Vertex stakeholders only.

1 Executive Summary

Apex Analytics gives revenue and operations teams live dashboards over their business metrics, but every insight today is *pull*: a user has to open the dashboard, notice a change, and interpret it themselves. Support tickets and win/loss interviews both point to the same failure mode — customers discover a metric drop (churned cohort, failed sync, revenue dip) days after it started, once the damage is already visible downstream.

Smart Alerts adds a push layer on top of the existing metrics pipeline: users subscribe a metric to anomaly monitoring, Apex's detection service watches it continuously, and a plain-language notification arrives in-app, by email, or in Slack the moment a statistically significant deviation is detected — before the user would have noticed on their own.

Why now. Enterprise renewal interviews (Feb 2026, $n = 22$) rank "proactive notification" as the #2 requested capability after SSO, and it is the single most-cited reason cited for evaluating competitor tools. Q3 is also when the anomaly-detection service built for internal ops dashboards (see ENG-2026-014) becomes available for customer-facing use, removing the main technical blocker.

Document Purpose

This PRD defines the problem, target users, scope, and success criteria for Smart Alerts v1, to align Product, Engineering, Design, and Data Science ahead of the Q3 2026 release planning cycle. It is the source of truth for scope decisions; implementation detail lives in the linked engineering RFC.

2 Problem Statement

The core problem

Apex customers only find out about metric anomalies when a human happens to look at the right chart at the right time. There is no mechanism today that watches a metric on a user's behalf and tells them when it moves outside its normal range.

Evidence

- Support: 34 tickets in the last 90 days include the phrase "we didn't notice until" — median detection lag of **4.2 days** after a metric anomaly began.
- Churn interviews (Feb 2026, $n = 22$): 14 of 22 accounts name "we found out too late" as a contributing factor to a near-miss or actual churn event.
- Product analytics: only 9% of weekly active users open a dashboard more than once per day; the other 91% are structurally unable to catch same-day anomalies under the current pull model.

Who feels this

- **Revenue operations leads** who own a pipeline or conversion metric and are accountable for explaining sudden drops to leadership, often after the fact.
- **Data-reliant engineering teams** who need to know within minutes when an upstream sync or ETL job silently breaks a metric feed, not when a downstream report looks wrong days later.
- **Customer success managers** who need early warning on account health metrics (usage, ticket volume) to intervene before a renewal conversation goes badly.

Why existing workarounds fail

Several accounts have built brittle stopgaps: scheduled CSV exports piped into a spreadsheet with manual threshold checks, or a Zapier automation polling the Apex API every hour and comparing to a hardcoded number. Both approaches break silently when a metric definition changes, produce false positives around normal seasonality (e.g. Monday traffic spikes), and require an engineer to maintain them. None generalize past the one metric they were built for.

3 Goals

Product goals

G1. Cut median anomaly-detection lag from 4.2 days to under 1 hour for subscribed metrics.

G2. Give every paid Apex workspace at least one active Smart Alert subscription within 30 days of GA, without requiring a sales-assisted setup call.

G3. Establish the anomaly-detection service as a reusable primitive other Apex surfaces (forecasting, reporting) can build on after v1.

Business goals

B1. Close the #2 enterprise renewal gap identified in Q1 interviews, supporting the Q3 enterprise renewal cohort ($1.4M ARR under active risk review).

B2. Create a differentiated feature for competitive deals against tools that only offer scheduled reports.

Non-goals (out of scope for v1)

- Alerting on custom SQL queries or ad-hoc metrics outside the standard metrics catalog.
- Predictive/forecast-based alerting ("this metric will breach target in 3 days"). Detection is on realized values only.
- Alert routing to third-party incident tools (PagerDuty, Opsgenie). Slack and email only for v1.
- Team-wide default subscriptions configured by an admin on behalf of others.

4 Target Users

Persona	Profile
Revenue Ops Lead (primary)	Owns 5–15 pipeline and conversion metrics across two or three dashboards. Checks Apex once or twice a day at fixed times; wants to be told, not to go looking. Success is never being surprised in a leadership review.
Data/Platform Engineer (primary)	Responsible for the data pipelines feeding Apex. Cares about detecting *feed* failures (a metric flatlining or going to zero) as much as business anomalies. Wants alerts routed to the team Slack channel, not personal email.
Customer Success Manager (secondary)	Tracks account health metrics for a book of 20–40 accounts. Needs early warning with enough context to act (which account, which metric, how unusual) without opening Apex first.
Workspace Admin (secondary)	Manages Apex billing and permissions. Cares about alert volume staying sane across the org and wants visibility into who is subscribed to what, but is not the primary subscriber.

5 User Stories

ID	User Story	Priority
US-01	As a **revenue ops lead**, I want to subscribe a dashboard metric to anomaly monitoring in two clicks, so that I don't have to configure thresholds manually.	P0
US-02	As a **revenue ops lead**, I want an alert to explain *how* unusual a change is (e.g. "38% below the 30-day expected range"), so that I can judge urgency without opening Apex.	P0
US-03	As a **data engineer**, I want alerts delivered to a shared Slack channel, so that the whole on-call rotation sees a feed failure at the same time.	P0
US-04	As a **data engineer**, I want to mute a metric during a known maintenance window, so that a planned migration doesn't page the team with false anomalies.	P1
US-05	As a **customer success manager**, I want an alert to include the account name and a one-line "why this matters" summary, so that I can triage without extra context-switching.	P1
US-06	As a **revenue ops lead**, I want to mark an alert as "expected" with one click, so that the detection model treats similar future changes as normal.	P1
US-07	As a **workspace admin**, I want a single view of every active alert subscription in my workspace, so that I can audit alert volume before it becomes noise.	P1
US-08	As a **revenue ops lead**, I want to set a sensitivity level (low / medium / high) per metric, so that a noisy metric doesn't drown out a stable one.	P2
US-09	As any **subscriber**, I want a weekly digest of all alerts I received, so that I can spot patterns without re-reading each individual notification.	P2

Priority key: P0 must ship for GA P1 ships within 4 weeks of GA P2 backlog, post-v1.

6 Scope

Release scope: v1 (Q3 2026)

✔ In scope

- Subscribe/unsubscribe to anomaly monitoring on any metric already in the Apex metrics catalog.
- Statistical anomaly detection (seasonal baseline + rolling standard deviation) on daily and hourly granularities.
- Delivery channels: in-app notification center, email, Slack (via existing Apex Slack integration).
- Per-metric sensitivity setting (low / medium / high).
- Maintenance-window muting.
- Mark-as-expected feedback loop that suppresses similar future alerts.
- Workspace-level subscription audit view for admins.

✖ Out of scope

- Custom SQL / ad-hoc metric alerting outside the standard catalog.
- Forecast-based ("will breach in N days") predictive alerting.
- PagerDuty, Opsgenie, or other third-party incident-tool routing.
- Admin-configured default subscriptions on behalf of other users.
- Mobile push notifications (mobile app does not yet exist).
- Alert-triggered automated actions (e.g. auto-rollback, auto-ticket creation).

Explicitly deferred to v2

Sub-minute detection latency, cross-metric correlation alerts ("these three metrics moved together"), and a public alerting API for customers to build their own integrations are all validated needs but are sequenced after v1 usage data confirms the core loop works.

7 Functional Requirements

ID	Requirement	Priority	Acceptance Criteria
FR-01	Users can subscribe any catalog metric to anomaly monitoring from the metric's detail view.	P0	Subscribe action visible on every catalog metric; confirmation toast on success; subscription visible in user's alert list within 2s.
FR-02	The detection service evaluates each subscribed metric on its native refresh cadence (hourly or daily) and flags values outside the expected range.	P0	95th-percentile detection-to-notification time under 5 minutes for hourly metrics, measured in staging load test.
FR-03	Alert notifications state the metric name, current value, expected range, and percentage deviation in plain language.	P0	Copy reviewed by Design; no raw z-scores or model internals shown to the end user.

. . . continued

ID	Requirement	Priority	Acceptance Criteria
FR-04	Users can choose delivery channel(s) per subscription: in-app, email, Slack.	P0	At least one channel required at subscribe time; channels editable after creation without re-subscribing.
FR-05	Users can set a maintenance window (start/end timestamp) during which a metric's alerts are suppressed but still logged.	P1	Suppressed alerts appear greyed out in history with a "muted" tag; no notification sent through any channel.
FR-06	Users can mark a received alert as "expected," which retrains the baseline to treat similar magnitude changes as normal going forward.	P1	Marking expected updates the model within 24h; verified via a repeat-alert suppression test.
FR-07	Workspace admins can view all active subscriptions across the workspace, filterable by metric and by user.	P1	Admin view loads under 2s for workspaces with up to 500 subscriptions.
FR-08	Users can set alert sensitivity (low / medium / high) per subscribed metric, changing the deviation threshold that triggers a notification.	P2	Threshold change takes effect on the next evaluation cycle; documented mapping from sensitivity level to standard-deviation multiplier.
FR-09	Subscribed users receive an optional weekly digest summarizing all alerts fired for their subscriptions.	P2	Digest opt-in default OFF; sent Monday 08:00 in the user's workspace timezone.

8 Success Metrics

North star

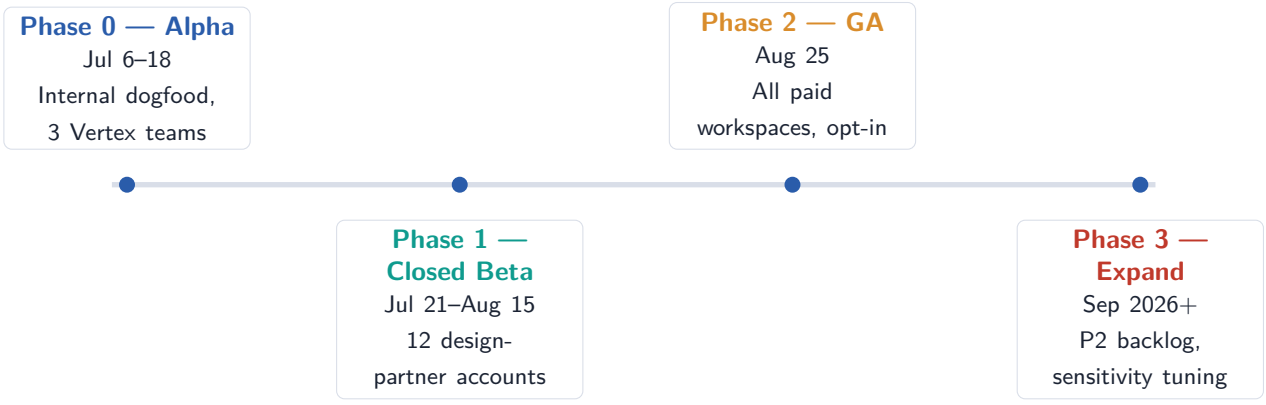
Median anomaly-detection lag
4.2 days → under 1 hour

Metric	Baseline	Target (90 days post-GA)	Measurement
Adoption: workspaces with $\geq$1 active subscription	0%	60% of paid workspaces	Weekly cohort query on `alert_subscriptions` table
Detection lag (subscribed metrics only)	4.2 days (proxy: ticket data)	$<$ 1 hour, p95	Instrumented end-to-end from anomaly timestamp to notification-sent event
Alert precision (share of alerts marked useful, not "expected" or dismissed)	n/a	$>$ 70%	In-product feedback on each alert (thumbs up / mark expected)
Alert fatigue: unsubscribes within 14 days of first alert	n/a	$<$ 10%	Subscription-lifecycle event log
Support tickets citing late anomaly discovery	34 / quarter	$<$ 15 / quarter	Support ticket tag `late-detection`, quarterly review
Enterprise renewal cohort retention (accounts citing alerting gap)	at risk: $1.4M ARR	$>$ 90% retained	CS pipeline tagging, reviewed at renewal

Guardrail metrics

Smart Alerts must not regress the metrics it depends on. Notification-center engagement and email open rates are tracked weekly during rollout; if Slack alert volume per workspace exceeds 15/day median, sensitivity defaults are revisited before wider rollout.

9 Release Plan



Phase exit criteria

Phase	Exit criteria
Alpha	Zero P0 bugs open; detection lag under 1 hour verified on at least 3 internal metrics with known synthetic anomalies.
Closed Beta	12 design partners onboarded with $\geq$1 subscription each; alert precision $\geq$ 60% measured from partner feedback; no more than 1 P1 bug open.
GA	Beta exit criteria hold at 4x the account count; on-call runbook signed off by Engineering; support macros published.
Expand	GA guardrail metrics stable for 4 consecutive weeks before P2 scope work begins.

Rollout mechanics

GA ships behind a workspace-level feature flag, enabled by default for new workspaces and opt-in via Settings for existing ones, to keep the Beta cohort's feedback loop intact while wider usage data accumulates.

10 Risks and Open Questions

Risk	Description	Mitigation
R1	Alert fatigue: if default sensitivity is too aggressive, users mute or unsubscribe before the feature proves value.	Ship conservative default sensitivity; surface "mark expected" prominently in the first alert a user receives.
R2	The anomaly-detection service (built for internal ops use) has not been load-tested against customer-facing traffic volumes.	Closed Beta explicitly load-tests at 4x design-partner volume before GA; Engineering owns a go/no-go checkpoint.
R3	Slack delivery depends on the existing Apex–Slack integration, which is independently owned and mid-refactor per ENG-2026-011.	Sync with Platform team before Beta; email remains a fallback channel if Slack delivery slips.
R4	Seasonal metrics (e.g. retail traffic) may produce false positives around known events (Black Friday) that the baseline model hasn't seen yet.	Ship a manual "pause during event" control as part of maintenance windows (FR-05) instead of waiting for seasonal model maturity.

Open questions

Q1. Should sensitivity default vary by metric type (revenue vs. operational), or should there be one global default for v1? *Owner: Omar Haddad, due before Beta kickoff.*

Q2. Do we need per-metric alert history retention limits, or does the standard 90-day Apex data retention policy apply unchanged? *Owner: Daniel Ruiz.*

Q3. Is a single "mark expected" click sufficient signal for the model, or does it need a reason code to avoid overcorrecting the baseline? *Owner: Data Science, resolve during Alpha.*

11 Appendix

Glossary

Anomaly	A metric value that falls outside its statistically expected range for the current time and seasonal context, as computed by the detection service.
Baseline	The expected-range model for a metric, built from its historical seasonal pattern and recent volatility.
Detection lag	Elapsed time between an anomaly's real-world onset and the moment a subscribed user is notified.
Sensitivity	A per-metric setting controlling how large a deviation must be before it triggers a notification.
Subscription	A user's opt-in link between a specific metric and one or more delivery channels.

Methodology note

The detection service's baseline model follows standard seasonal-trend decomposition practice [2], with anomaly scoring adapted from the generalized ESD approach used in production anomaly-detection systems at web scale [3]. Classical ARIMA-family forecasting [1] was evaluated during early spikes but rejected for v1 due to per-metric retraining cost; it remains a candidate for the Phase 3 forecasting extension noted in Section 3.

Revision history

Version	Date	Change
1.0	Feb 2, 2026	Initial draft circulated to Engineering and Design leads.
1.1	Feb 18, 2026	Added maintenance-window muting (FR-05) after Beta-partner feedback session.
1.2	Mar 3, 2026	Split success metrics into north star + guardrails; added enterprise renewal ARR context.
1.3	Mar 9, 2026	Finalized release plan phases and exit criteria for review.

References

[1] George E. P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970.

[2] Robert B. Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73, 1990.

[3] Jordan Hochenbaum, Owen S. Vallis, and Arun Kejariwal. Automatic anomaly detection in the cloud via statistical learning. In *arXiv preprint arXiv:1704.07706*, 2017.