Handz V1 — Chapter CH33

```
Doc ID: HZ-V1-CH33_Analytics_and_Metrics_Spec_R1
Revision: R1 (2026-01-02)
Status: Draft
Depends on: CH01, CH02, CH03, CH04, CH05, CH06, CH07, CH08, CH12, CH15, CH16, CH17, CH18,
CH20, CH21, CH22, CH23, CH24, CH25, CH30, CH31, CH34
Related: CH26, CH27, CH28, CH29, CH35, CH36, CH37, CH38
Supersedes: N/A
Owned Decisions: Analytics event naming + schema; required event properties; funnel
definitions; dashboard minimum set; privacy-by-default analytics rules; analytics QA
checklist for V1
Open Questions / Placeholders: (see §14)
```

0. Scope and intent

This chapter defines exactly what Handz measures in V1, how those measurements are structured, and how analytics must respect privacy. It is written so implementation can be done with minimal guessing while still allowing future evolution (new events, new dashboards, new experiments).

- Audience: product + engineering. This is a build-spec, not marketing copy.
- Primary output: a canonical event catalog + funnel definitions + dashboard requirements.
- Non-goal: picking a specific analytics vendor (placeholder).
- Rule: when another chapter owns the product rule, do not re-define it here. Cross-reference it. Example: See: HZ-V1-CH30 §2 (Warning Ladder).

1. Analytics principles (must-follow)

- Minimum necessary data
- Collect only what is needed to ship/iterate V1 responsibly: adoption, engagement, paywall conversion, reliability, and abuse signals.
- Default posture: no PII; no content text; no raw media; no opponent names; no free-form notes.
- Privacy-by-default
- No advertising tracking; no cross-app tracking; no IDFA usage in V1.
- Analytics must be first-party: event data is used only to improve Handz.
- Provide a user-facing toggle to disable analytics collection (placeholder decision on default).
- Deterministic identifiers
- Use stable internal IDs (flow_id, move_id, path_id, gameplan_id) so we can measure outcomes without storing user content.
- Identifiers must be random UUIDs; never derived from user text.
- Event catalog is authoritative
- Every event must be listed in §6 or explicitly marked as future/placeholder.
- No “random logging” in code. If needed during debugging, use local dev logs that never ship to production.

- “Attempt” events are required
- Whenever a user hits a gate (Guest restriction, Free cap, Pro-only feature), emit BOTH a gate_attempt event and a gate_result event (blocked/allowed).
- This prevents silent funnels where we don’t know why users drop.

2. Privacy, compliance, and user controls

This section defines what analytics is allowed to collect, how consent/controls work, and how we avoid trust-breaking behaviors.

2.1 Data categories (explicit allow/deny list)

- Allowed (V1)
- Product usage events: screens visited, button taps, feature usage, counts, durations.
- Structural metrics: number of flows, nodes, branches; practice session timing; credits usage.
- Plan state + entitlement state (Guest/Free/Trial/Pro). See: HZ-V1-CH08 §1 (Plan States).
- Performance telemetry: load time buckets, canvas FPS bucket, crash-free sessions.
- Error codes and non-sensitive diagnostics (no raw stack traces unless redacted).
- Not allowed (V1)
- Any user-entered free-form text (notes, custom move descriptions, sequence detail text).
- Exact flow graphs or path sequences (we can measure counts/shape, not the content).
- Media files, thumbnails, or URLs beyond domain-level classification (e.g., “youtube.com”).
- Contacts, address book, precise location, microphone audio, camera footage.
- Advertising identifiers (IDFA) or third-party trackers.

2.2 User-facing settings + defaults

- Settings > Privacy:
- Analytics toggle
- Label: “Share anonymous usage data to improve Handz”.
- If OFF: stop sending analytics events; keep local-only counters needed for on-device UX (e.g., streak display) but do not upload.
- If user turns ON later: resume sending events; do NOT backfill historical events by default (placeholder option).
- Crash reporting toggle
- Label: “Share crash reports”.
- May be separate from analytics. If OFF: do not send crash logs; still show user-friendly error messages.
- Data export + delete
- Must link to CH34 flows: export and deletion also affects analytics identifiers.
- Default policy

- PLACEHOLDER: Default analytics toggle state (On vs Off) • Owner: CH33 • Options: A) On by default with clear disclosure; B) Off by default; C) Ask at first launch.
- Recommendation for V1: A) On by default with clear disclosure and easy opt-out, unless legal counsel requires otherwise.

2.3 Consent surfaces (where we disclose)

- First launch: short disclosure in onboarding (non-blocking).
- Settings > Privacy: full details + toggle(s).
- Privacy Policy link in Settings and during signup. (Copy owned by CH07/CH08/CH34; we reference it here.)

3. Core analytics objects + identifiers

To avoid tracking content while still measuring outcomes, analytics relies on stable IDs and structured metadata.

3.1 Identifier glossary (analytics-facing)

- install_id: Random UUID created at first app launch. Resets on reinstall.
- user_id: Random UUID assigned after account creation. Never email-based.
- session_id: Random UUID per app session (foreground start until background > threshold).
- flow_id: UUID per saved flow (library).
- flow_draft_id: UUID per unsaved draft (if supported; see CH28/CH15).
- move_id: UUID for canonical/default move OR user custom move. (Canonical move also has canonical_id; see CH10).
- path_id: UUID for a selected path instance (usually derived from flow edge traversal).
- gameplan_id: UUID for a user-defined bundle of paths/flows (see CH23).
- share_link_id: UUID for an unlisted link instance (see CH17).
- import_id: UUID for an inbox item (see CH18).
- practice_session_id: UUID for a practice session (see CH20–CH22).
- mastery_plan_id: UUID for a mastery plan (see CH23).
- maintenance_plan_id: UUID for maintenance schedule bundle (see CH24).

3.2 Global event properties (required on every event)

Every analytics event MUST include these base properties:

Property	Type	Notes
event_name	string	snake_case name, must exist in §6
event_time_utc	ISO-8601	device time in UTC
app_version	string	marketing version
build_number	string/int	internal build
platform	string	iOS
os_version	string	e.g., iOS 18.6
device_model	string	iPhone model string
locale	string	e.g., en_US
timezone	string	IANA tz
session_id	uuid	current session
install_id	uuid	install-scoped
user_state	enum	guest | free | trial | pro
analytics_opt_in	bool	true/false at time of event
network_state	enum	online | offline | flaky

- Optional common properties (include when relevant)
- screen_name (string): current screen route identifier (See: HZ-V1-CH04 §2).
- entry_point (enum/string): where the action started (e.g., library, share_link, inbox).
- duration_ms (int): for timed actions (load, practice, render).
- error_code (string): for errors (See: HZ-V1-CH31 §1).
- gate_name (string): for paywalls/caps (See: HZ-V1-CH08 §3).

4. Event naming standard + categories

- Format: _[_] in snake_case.
- Domain examples: app, auth, onboarding, moves, flows, builder, share, inbox, practice, mastery, maintenance, paywall, settings, error, perf.
- Action is a verb: viewed, tapped, created, saved, started, completed, failed, blocked, allowed, exported.
- Do not include user content in names or properties.
- If an event is emitted from multiple screens, include screen_name property rather than duplicating event names.

4.1 Domains (authoritative list for V1)

- app, auth, onboarding, nav, moves, flows, builder, sequence, library, share, inbox, practice, mastery, maintenance, paywall, settings, notif, export, error, perf, abuse

5. Funnels and KPIs (what success looks like)

Funnels are defined as ordered event sequences. Each funnel must be measurable end-to-end.

5.1 Core funnels (V1)

- F0 — Activation (first value)
- Start: app_open
- Step 1: onboarding_viewed
- Step 2: flow_builder_opened (or library_viewed if starting there)
- Step 3: flow_draft_created
- Step 4: flow_saved_success
- Success: user has at least 1 saved flow.
- F1 — Account conversion
- Start: guest_mode_entered
- Step: guest_restriction_hit (any)
- Step: signup_viewed
- Step: signup_success

- Success: user_id created + user_state becomes free.
- F2 — Sharing growth loop (unlisted links)
- Start: share_link_create_tapped
- Step: share_link_created
- Step: share_link_opened (by receiver)
- Step: import_received
- Step: import_saved_to_library (receiver)
- Success: receiver now has a saved flow and later creates their own.
- F3 — Practice paywall conversion
- Start: practice_entry_attempt
- Step: paywall_viewed (if blocked)
- Step: trial_started or subscription_purchased
- Step: practice_session_started
- Success: first completed practice session after upgrade.
- F4 — Mastery plan adoption
- Start: mastery_plan_created
- Step: mastery_plan_scheduled
- Step: mastery_practice_started
- Success: plan has ongoing maintenance activity.

5.2 KPI definitions (minimum set)

- Activation
- Activation rate: % installs that reach flow_saved_success within 24h / 7d.
- Time-to-first-flow: median time from app_open to flow_saved_success.
- Retention
- D1/D7/D30 retention: % of users returning on day N after install.
- Practice retention: % of users who run practice at least once per week.
- Engagement
- Weekly flows edited per active user.
- Weekly practice minutes per practicing user (actual duration; See: HZ-V1-CH22).
- Branches-per-flow distribution (complexity appetite).
- Monetization
- Paywall view rate: % of users seeing paywall per week.
- Trial start rate and trial-to-paid conversion.
- Churn rate: monthly subscription cancellations.
- Sharing loop

- Share link create rate per active user.
- Share link open-to-import rate.
- Import-to-saved conversion rate.
- Quality
- Crash-free sessions percentage.
- Canvas performance: % sessions with builder_fps_bucket=good.

6. Canonical event catalog (V1)

This is the authoritative list of analytics events for V1. If it’s not listed here, it must not ship (unless explicitly marked as a placeholder and disabled by default).

6.1 App + session

app_open

Triggered when: App enters foreground and becomes active (cold start or warm start).

Properties:

- app_open_reason (enum): cold_start | warm_start | push_notification | deep_link
- deep_link_type (enum|null): share_link | other (future)
- time_since_install_s (int|null)

Notes:

- Emit once per foreground session.

app_background

Triggered when: App goes to background (or is terminated).

Properties:

- foreground_duration_ms (int)
- open_screens_count (int)

Notes:

- Used to compute session length distributions.

screen_viewed

Triggered when: A screen becomes visible (route change).

Properties:

- screen_name (string)
- previous_screen_name (string|null)
- entry_point (string|null)

Notes:

- This can be auto-instrumented in navigation layer.
- Do not log every minor UI panel—only real screens/routes.

6.2 Onboarding + education

onboarding_viewed

Triggered when: User sees the onboarding start screen.

Properties:

- variant (enum): minimal_a | minimal_b (placeholder for experimentation)
- user_state

onboarding_completed

Triggered when: User finishes onboarding (reaches first dashboard/library).

Properties:

- completion_path (enum): guest | signup
- duration_ms (int)

education_flow_concept_opened

Triggered when: User opens the “What is a flow?” explainer (wherever it lives; see CH04/CH05).

Properties:

- entry_point (string)
- time_in_view_ms (int|null)

Notes:

- This is critical because the concept is unfamiliar for many users; we must measure comprehension surfaces.

6.3 Authentication + account conversion

guest_mode_entered

Triggered when: User starts the app in Guest mode.

Properties:

- entry_point (enum): first_launch | logout | skip_signup

signup_viewed

Triggered when: User views the signup screen.

Properties:

- entry_point (string)
- reason (string|null): guest_restriction | paywall | voluntary

signup_method_selected

Triggered when: User selects a signup method.

Properties:

- method (enum): apple | google | email
- entry_point (string)

signup_success

Triggered when: Signup completes successfully and a user_id is created.

Properties:

- method (enum): apple | google | email
- duration_ms (int)
- previous_user_state (enum): guest

signup_failed

Triggered when: Signup attempt fails.

Properties:

- method (enum)
- error_code (string)
- error_step (string)
- duration_ms (int|null)

Notes:

- Do not log raw provider error messages if they may contain PII.

login_success

Triggered when: Existing user logs in successfully.

Properties:

- method (enum): apple | google | email
- duration_ms (int)

6.4 Moves (default + custom)

moves_library_viewed

Triggered when: User views the moves picker/library UI (during onboarding or later).

Properties:

- entry_point (string)
- mode (enum): onboarding | settings | builder_add_move

move_selected

Triggered when: User selects a move from the default list for use or inclusion.

Properties:

- move_canonical_id (string)
- context (enum): onboarding_pack | builder_add | filter_select

custom_move_create_started

Triggered when: User begins creating a custom move (see CH11).

Properties:

- entry_point (string)

custom_move_create_saved

Triggered when: User saves a custom move.

Properties:

- move_id (uuid)
- has_media_link (bool)
- has_media_upload (bool)
- alias_count (int)
- tag_count (int)
- family_assigned (bool)
- variant_of_move_id (uuid|null)

Notes:

- Do not send the move name, aliases, or notes; only counts and booleans.

custom_move_edit_saved

Triggered when: User edits and saves an existing custom move.

Properties:

- move_id (uuid)
- edit_type (enum): rename | aliases | tags | family | media | other
- revert_used (bool)

move_revert_used

Triggered when: User uses revert for a move or move attribute (see CH11).

Properties:

- target (enum): move | field
- field_name (string|null)
- scope (enum): this_move | library

6.5 Flow builder (creation + editing)

flow_builder_opened

Triggered when: User opens the flow builder canvas.

Properties:

- entry_point (enum): new_flow | edit_flow | from_template | from_inbox
- flow_id (uuid|null)
- flow_draft_id (uuid|null)

flow_draft_created

Triggered when: User creates a new flow draft (unsaved).

Properties:

- flow_draft_id (uuid)
- template_used (bool)
- template_source (enum|null): share_link | inbox | builtin_demo

builder_node_added

Triggered when: User adds a node to the canvas.

Properties:

- node_type (enum): move | sequence | label
- move_canonical_id (string|null)
- move_id (uuid|null)
- via (enum): picker | quick_add | duplicate
- total_nodes_after (int)

builder_edge_created

Triggered when: User creates a connection (edge) between nodes.

Properties:

- edge_type (enum): normal | branch
- from_node_type (enum)
- to_node_type (enum)
- branch_index (int|null)
- total_outgoing_from_source_after (int)

builder_branch_limit_hit

Triggered when: User attempts to add a branch beyond the allowed max (10). See: HZ-V1-CH12 §3.4.

Properties:

- flow_id (uuid|null)
- flow_draft_id (uuid|null)
- source_node_id (uuid)
- attempted_outgoing_count (int)

Notes:

- This should also emit gate_attempt/gate_result with gate_name=branch_limit.

builder_node_reordered

Triggered when: User reorders nodes or changes layout order (if supported).

Properties:

- method (enum): drag_reorder | auto_layout
- nodes_moved_count (int)

builder_root_replaced

Triggered when: User replaces the root move of a flow.

Properties:

- previous_root_move_id (uuid|null)
- new_root_move_id (uuid|null)
- new_root_canonical_id (string|null)

sequence_editor_opened

Triggered when: User opens the sequence detail editor for a connection. See: HZ-V1-CH14.

Properties:

- entry_point (enum): edge_tap | node_context_menu
- has_existing_details (bool)

sequence_editor_saved

Triggered when: User saves sequence details between two nodes.

Properties:

- fields_used_count (int)
- has_media_link (bool)
- has_media_upload (bool)

flow_save_attempt

Triggered when: User attempts to save a flow to their library.

Properties:

- user_state
- flow_draft_id (uuid)
- current_saved_flow_count (int|null)

flow_saved_success

Triggered when: Flow is saved successfully.

Properties:

- flow_id (uuid)
- nodes_count (int)

- edges_count (int)
- branches_count (int)
- dangling_paths_count (int)
- sequence_nodes_count (int)
- duration_ms (int)

flow_saved_blocked

Triggered when: Flow save is blocked (Guest restriction or Free cap).

Properties:

- gate_name (enum): guest_save_required | free_saved_flows_cap
- current_saved_flow_count (int|null)

Notes:

- See: HZ-V1-CH08 §3 (Entitlement rules).
- Also emit gate_attempt and gate_result.

6.6 Library + organization

library_viewed

Triggered when: User views the Library tab/main screen.

Properties:

- tab (string): library
- flows_count (int|null)
- folders_count (int|null)

folder_created

Triggered when: User creates a folder.

Properties:

- folder_id (uuid)
- parent_folder_id (uuid|null)

flow_opened

Triggered when: User opens a flow detail view (not builder). See: HZ-V1-CH16.

Properties:

- flow_id (uuid)
- entry_point (enum): library | search | folder | import_saved

flow_duplicated

Triggered when: User duplicates a flow into their library.

Properties:

- source (enum): library | share_link | inbox
- source_flow_id (uuid|null)
- new_flow_id (uuid)
- user_state

6.7 Sharing + imports

share_sheet_opened

Triggered when: User opens sharing UI for a flow.

Properties:

- flow_id (uuid)
- entry_point (enum): flow_detail | builder

share_link_create_tapped

Triggered when: User taps “Create unlisted link”.

Properties:

- flow_id (uuid)
- current_links_created_today (int|null)
- user_state

share_link_created

Triggered when: Unlisted share link is created successfully. See: HZ-V1-CH17.

Properties:

- share_link_id (uuid)
- flow_id (uuid)
- expires_in_days (int|null)
- user_state

share_link_create_blocked

Triggered when: Share link creation is blocked (rate limit or plan).

Properties:

- gate_name (enum): share_link_rate_limit | plan_required
- current_links_created_today (int|null)

Notes:

- See: HZ-V1-CH17 §3 (Link limits) and CH30 (abuse thresholds).

share_link_opened

Triggered when: A recipient opens an unlisted link.

Properties:

- share_link_id (uuid)
- entry_point (enum): deep_link
- is_logged_in (bool)

import_received

Triggered when: A link open results in an inbox item being created for the recipient.

Properties:

- import_id (uuid)
- share_link_id (uuid)
- recipient_user_state (enum): guest | free | trial | pro

inbox_viewed

Triggered when: User views inbox list. See: HZ-V1-CH18.

Properties:

- inbox_count (int)
- user_state

import_opened

Triggered when: User opens an inbox item detail view.

Properties:

- import_id (uuid)
- contains_custom_moves (bool)
- contains_custom_media_uploads (bool)
- contains_custom_media_links (bool)

import_saved_to_library_attempt

Triggered when: User attempts to save an imported flow into their library.

Properties:

- import_id (uuid)
- user_state
- current_saved_flow_count (int|null)
- inbox_count (int|null)

import_saved_to_library_success

Triggered when: Imported flow is saved into library successfully.

Properties:

- import_id (uuid)
- new_flow_id (uuid)
- conflict_resolution_used (bool)

import_saved_to_library_blocked

Triggered when: Import save blocked due to Free saved flow cap or other gate.

Properties:

- gate_name (enum): free_saved_flows_cap | account_required
- current_saved_flow_count (int|null)

6.8 Practice (drills) + credits

practice_entry_attempt

Triggered when: User attempts to enter Practice for a given saved flow/gameplan.

Properties:

- entry_point (enum): flow_detail | gameplan_detail | home_recommendation
- user_state
- target_type (enum): flow | gameplan
- target_id (uuid)

practice_gate_result

Triggered when: Result of gating when entering Practice.

Properties:

- gate_name (enum): practice_pro_required | trial_required | credits_required | none
- result (enum): allowed | blocked
- user_state
- credits_remaining (int|null)

practice_session_started

Triggered when: User starts an actual practice session (timer begins).

Properties:

- practice_session_id (uuid)
- target_type (enum): flow | gameplan
- target_id (uuid)
- selected_paths_count (int)
- timer_seconds (int)
- assumed_reps (int)
- mode (enum): drill | demo

practice_path_presented

Triggered when: A path step is presented to the user during practice.

Properties:

- practice_session_id (uuid)
- path_id (uuid)
- step_index (int)
- total_steps (int)

Notes:

- Do not send actual move names or sequence text.

practice_session_interrupted

Triggered when: Practice session is interrupted and saved as interrupted.

Properties:

- practice_session_id (uuid)
- reason (enum): user_exit | app_background | call_interrupt | timeout
- elapsed_seconds (int)
- completed_steps (int)

practice_session_completed

Triggered when: Practice session completes (timer finishes or user taps Completed).

Properties:

- practice_session_id (uuid)
- completion_type (enum): timer_finished | user_completed
- actual_duration_seconds (int)
- completed_steps (int)
- selected_paths_count (int)

practice_credit_spent

Triggered when: A practice credit is spent (free monthly credits).

Properties:

- credit_type (enum): monthly_free
- credits_remaining_after (int)
- source (enum): practice_start

practice_credit_refreshed

Triggered when: Monthly free credits refresh.

Properties:

- credits_added (int)
- credits_total_after (int)
- user_state (enum): free

practice_demo_started

Triggered when: User starts the built-in demo practice (predefined content).

Properties:

- practice_session_id (uuid)
- demo_id (string)
- timer_seconds (int)

6.9 Mastery + maintenance planning

gameplan_created

Triggered when: User creates a gameplan (bundle of paths/flows). See: HZ-V1-CH23.

Properties:

- gameplan_id (uuid)
- source (enum): flow | multi_flow | manual
- items_count (int)

mastery_plan_created

Triggered when: User creates a mastery plan from a path/flow/gameplan.

Properties:

- mastery_plan_id (uuid)
- target_type (enum): path | flow | gameplan
- target_id (uuid)
- goal_type (enum): recall | combat_readiness
- schedule_days_per_week (int)
- sessions_per_day (int)

mastery_plan_updated

Triggered when: User updates mastery plan parameters.

Properties:

- mastery_plan_id (uuid)
- fields_changed_count (int)
- maintenance_enabled (bool)

maintenance_plan_created

Triggered when: User creates or enables a maintenance plan.

Properties:

- maintenance_plan_id (uuid)
- items_count (int)
- daily_budget_minutes (int)
- notif_enabled (bool)

maintenance_overload_warning_shown

Triggered when: User is warned about maintenance overload risk (too many items).

Properties:

- maintenance_plan_id (uuid)
- items_count (int)
- daily_budget_minutes (int)

maintenance_session_completed

Triggered when: User completes a maintenance drill session.

Properties:

- maintenance_plan_id (uuid)
- actual_duration_seconds (int)
- items_drilled_count (int)

6.10 Paywall + subscription

paywall_viewed

Triggered when: User views a paywall screen.

Properties:

- paywall_id (enum): practice | saved_flows | sharing (future) | other
- entry_point (string)
- user_state

paywall_cta_tapped

Triggered when: User taps primary CTA on paywall.

Properties:

- paywall_id (enum)
- cta (enum): start_trial | subscribe | restore | not_now

trial_started

Triggered when: User starts a trial (7 days).

Properties:

- plan (string): monthly_9_99
- trial_days (int): 7

subscription_purchased

Triggered when: User completes a purchase.

Properties:

- plan (string): monthly_9_99
- intro_offer (bool)
- via (enum): paywall | settings

subscription_cancelled

Triggered when: User cancels subscription (if detectable via StoreKit status change).

Properties:

- plan (string)
- days_since_purchase (int|null)

restore_purchases_success

Triggered when: User restores purchases successfully.

Properties:

- plan (string)

restore_purchases_failed

Triggered when: Restore purchases fails.

Properties:

- error_code (string)

6.11 Errors + performance

error_shown

Triggered when: A user-visible error state is shown. See: HZ-V1-CH31.

Properties:

- error_code (string)
- screen_name (string)
- severity (enum): info | warning | blocking

perf_flow_builder_loaded

Triggered when: Flow builder fully loads and is interactive.

Properties:

- duration_ms (int)
- nodes_count (int|null)
- edges_count (int|null)
- fps_bucket (enum): good | ok | poor

perf_flow_render_time

Triggered when: Flow graph render/relayout time measured (bucketed).

Properties:

- duration_ms (int)
- nodes_count (int)
- branches_count (int)
- layout_mode (enum): vertical | horizontal

crash_reported

Triggered when: A crash report is submitted (if enabled).

Properties:

- last_screen_name (string|null)
- had_unsaved_changes (bool)
- flow_draft_id (uuid|null)

6.12 Abuse + safety signals (analytics-only)

abuse_rate_limit_triggered

Triggered when: A user hits a safety rate limit (share creation, login attempts, uploads). See: HZ-V1-CH30.

Properties:

- limit_name (string)
- window (string)
- attempts_in_window (int)
- action_blocked (bool)

abuse_warning_ladder_step

Triggered when: User is advanced on the warning ladder.

Properties:

- ladder_step (int)
- reason_code (string)
- cooldown_hours (int|null)

7. Dashboards and reporting requirements

Dashboards define what the team must be able to see on day 1 of launch. They are expressed as queries over the event catalog above.

7.1 Dashboard: Launch health (daily)

- Crash-free sessions % (last 24h, 7d rolling).
- App_open count, unique users, sessions.
- Error_shown by error_code (top 10).
- Perf_flow_builder_loaded: p50/p90 duration_ms, fps_bucket distribution.
- Signup_success rate and signup_failed by method/error_code.

7.2 Dashboard: Activation + onboarding

- F0 activation conversion (app_open -> flow_saved_success) with drop-off per step.
- Median time-to-first-flow.
- Education_flow_concept_opened rate; correlation with activation.
- Guest_mode_entered -> signup_success funnel (F1).

7.3 Dashboard: Builder usage + complexity

- Flows created per day and cumulative.
- Distribution: nodes_count, edges_count, branches_count, dangling_paths_count.
- Builder_branch_limit_hit count (indicates users want more than 10 branches).
- Sequence_editor_saved rate (are users using details?).

7.4 Dashboard: Sharing loop

- share_link_created per day; by user_state.
- share_link_opened -> import_received -> import_saved_to_library_success conversion.
- Inbox cap hits (free inbox cap=10) and resulting signup/upgrade behavior.

7.5 Dashboard: Practice + mastery

- practice_entry_attempt and practice_gate_result breakdown (allowed vs blocked).
- practice_session_started and practice_session_completed counts, completion rate.
- Practice duration distribution; interruption reasons.
- Credits: practice_credit_spent distribution; % users exhausting credits.
- Mastery/maintenance adoption: mastery_plan_created, maintenance_plan_created.

7.6 Dashboard: Monetization

- paywall_viewed by paywall_id and entry_point.
- paywall_cta_tapped rates; start_trial vs not_now.

- trial_started count and trial-to-paid conversion (subscription_purchased after trial).
- Churn (subscription_cancelled) over time.

8. Data quality, QA, and validation

Analytics is only useful if it is consistent and trustworthy. This section defines the validation rules and QA checklist.

8.1 Validation rules

- All events must include base properties (§3.2). Reject events missing any required base property (except event_time_utc in rare offline cases; see §9).
- All event_name values must match a known name from §6. Unknown events must be blocked in production builds.
- Property types must match the catalog. If a property is optional, it must be omitted rather than sent as garbage values.
- IDs must be UUID format strings where specified.
- No PII scanning: build-time lint that rejects suspicious keys (email, phone, name, notes, description).

8.2 QA checklist (pre-release)

- Cold start: app_open emits exactly once; screen_viewed emits on first screen.
- Guest flow: guest_mode_entered -> guest restriction -> signup funnel emits expected events.
- Save flow blocked: flow_save_attempt + flow_saved_blocked + gate_attempt/gate_result emitted.
- Share link: share_link_create_tapped -> share_link_created; open on another device yields share_link_opened + import_received.
- Inbox cap: at 10 items, new import emits import_received + import_save_blocked or inbox_cap gate event (exact behavior per CH18).
- Practice: free user blocked emits practice_entry_attempt + practice_gate_result(blocked). Pro user emits practice_session_started and completion events.
- Credits: practice_credit_spent decrements correctly; refresh emits practice_credit_refreshed monthly.
- Paywall: paywall_viewed -> paywall_cta_tapped -> trial_started/subscription_purchased; restore flows log restore events.
- Performance: perf_flow_builder_loaded measures duration and buckets.
- Analytics opt-out: toggling OFF stops all event sending immediately.

9. Offline behavior for analytics

Handz must work offline for core viewing/editing. Analytics must not break offline behavior and must not leak data.

- Offline queue

- If analytics is enabled, events generated offline are queued locally until network resumes.
- Queue size cap (placeholder): recommended 1,000 events or 5 MB, whichever comes first.
- When cap is reached: drop oldest events and emit a local-only counter (do NOT upload a special event to avoid recursion).
- Timestamp rules
- Each event carries device event_time_utc.
- When uploaded, backend may add received_time_utc for latency analysis.
- User logout/account delete
- If user logs out: stop sending user_id. Continue install_id based analytics unless analytics toggle is off.
- If user deletes account: delete user_id mapping; future events are anonymous. See CH34 for data deletion guarantees.

10. Security and abuse considerations for analytics

- Analytics endpoints must be authenticated or use signed ingestion keys to prevent spoofing.
- Do not accept arbitrary event names from client in a way that could be exploited (use allowlist).
- Rate-limit analytics ingestion to prevent DoS from compromised clients.
- Store analytics data with least privilege; restrict internal access.
- Redact sensitive crash logs; disable logs in release unless opted-in.

11. Experimentation readiness (future-proof, optional in V1)

- Feature flags
- Ability to enable/disable certain features remotely (e.g., onboarding variants).
- Analytics must log flag_state_snapshot (optional property) when relevant so results can be segmented.
- A/B tests
- If running tests in V1, define experiment_id and variant in event properties (optional).
- Do not run more than 1–2 simultaneous tests at launch to avoid data confusion.

12. Troubleshooting and debugging analytics

- Developer diagnostics screen (hidden)
- Screen shows: analytics enabled state, queued events count, last 10 events, last upload result code, last error.
- Accessible via hidden gesture in Settings (e.g., tap version number 7 times).
- Common issues and what to check
- No events arriving: confirm analytics toggle ON, network_state, ingestion key, allowlist.
- Duplicate events: confirm app_open emit logic (only once per foreground) and navigation screen_viewed debouncing.
- Wrong plan segmentation: confirm user_state property is updated immediately after purchase restore.
- Weird funnel drop-offs: verify gate_attempt/gate_result are implemented for every cap and paywall.

13. Acceptance tests (chapter complete definition)

- Event allowlist implemented and matches §6 exactly.
- All screens emit screen_viewed with correct screen_name and previous_screen_name.
- All major funnels (F0–F4) can be computed from events without missing steps.
- Analytics opt-out fully stops analytics network calls.
- No PII leakage confirmed by automated lint + manual review.
- Offline queue works and never crashes the app; queue cap enforced.
- Dashboards in §7 can be created with available events/properties.

14. Placeholders / decisions to finalize later

- Analytics vendor
- Options: Firebase Analytics, PostHog, Amplitude, Mixpanel, custom.
- Constraint: iOS-only V1; must support privacy-friendly configuration.
- Decision owner: Product/Engineering. Default suggestion for speed: Firebase Analytics + Crashlytics (if allowed).
- Default analytics opt-in policy

- Choose A/B/C from §2.2 based on legal + launch risk tolerance.
- Event retention period
- Placeholder: 13 months vs 24 months vs 36 months.
- Note: Longer retention improves cohort analysis but increases data liability.
- Sampling
- Default: no sampling for core funnels; optional sampling for high-frequency perf events.
- Define sampling rate if needed (e.g., 10% for perf_flow_render_time).
- Device identifier hash usage
- If needed for abuse prevention: use salted rotating hash; never store raw device identifiers.
- If not needed: omit entirely.
- Experiment system
- Remote config and A/B testing can be deferred until after launch.

15. Replit build prompt (drop-in)

```
You are building Handz (iOS-only) per the PRD chapter HZ-V1-CH33 (Analytics & Metrics).
Implement analytics instrumentation as follows:

1) Create an event allowlist matching exactly the event names in CH33 §6. Unknown events
must be blocked in production builds.
2) Every event must include required base properties from CH33 §3.2.
3) Add automatic screen_viewed tracking for every top-level route/screen.
4) Implement gating attempt/result events for: guest_save_required, free_saved_flows_cap,
practice_pro_required, inbox_cap, share_link_rate_limit, branch_limit.
5) Implement offline event queue (cap 1000 events or 5MB). Queue flushes when network
returns.
6) Add Settings > Privacy toggles: analytics enabled, crash reporting enabled. Turning off
analytics stops event sending immediately.
7) Add a hidden developer diagnostics screen showing analytics status and last 10 events.
8) Ensure no PII is ever logged. Do not log move names, flow titles, notes, descriptions,
media URLs beyond domain classification, or any free-form text.

Output:
- analytics module/library wrapper
- event schemas/types
- unit tests or simple runtime asserts for allowlist + required properties
- documentation in-code that references CH33 sections for each part
```

16. Troubleshooting checklist for you (non-coder friendly)

- If events aren’t showing up
- Check Settings > Privacy: analytics toggle must be ON.
- Check the developer diagnostics screen: queued events count should change when you tap around.
- If queued events keep growing: you may be offline or ingestion config is wrong.
- If funnels look wrong
- Confirm you are testing with a clean install and known user_state (Guest vs Free vs Pro).

- Confirm that blocked actions emit both the attempt and the blocked result events (flow_saved_blocked, practice_gate_result, etc.).
- If you see sensitive data in logs
- Stop shipping immediately and remove the property/event.
- Update the PII lint denylist to catch the key in the future.