Technical report v1.0 2 November 2026 33 min read

Beyond the Default: A Coded Study of 242 Design Corrections One Designer Gave AI Coding Agents Across 15 Products

Yousof Selim
Sunway University (Lancaster University dual degree), Malaysia

71% of design corrections came after a rule existed

1. 173 of 242 design corrections (71%) came after a written rule for that kind of tell already existed; for 92 of them the rule existed only in another product's repository.
2. There was no sign that a rule lowered the correction rate within a product: 0.35 corrections per review day before a rule and 0.38 after, across 63 product and tell pairs (rate ratio 1.08, rough 95% CI 0.80 to 1.47; observational and confounded).
3. Type corrections rose from 4% of corrections in June and July (3 of 79) to 12% in October and 1 November (19 of 153); the wider shift from too-much to too-safe complaints (48% to 59% too safe) was not statistically distinguishable from chance.
4. All 15 kinds of AI-default tell recurred in at least three products, and screenshot or device review was the steering method seen most often, in 59 corrections.

Limits. One designer's self-recorded corrections over five months, screened and coded by an AI assistant with no second human coder; rules are matched by tell category, the before-and-after comparison is confounded, and his taste moved during the study.

Abstract

The problem	AI coding agents build working interfaces that often look generic. Designers who direct them keep correcting the same habits, and the usual remedy is to write the rules down in files the agents read.
What I did	I coded 242 design corrections I gave Claude Code and Codex across 15 of my products from June to November 2026, using my own messages, agent memory notes, design documents and commits. I linked each correction to the written rules that existed at the time. An AI assistant did the screening and coding; there was no second human coder.
What I found	**71% of corrections (173 of 242) came after a rule for that kind of tell was already written, 92 of them only in another product. Correction rates were no lower after a rule (0.38 per review day) than before (0.35). Type complaints rose from 4% to 12% of corrections.**
Why it matters	In this record, written design rules recorded taste but did not, on their own, keep an agent's look on track, and rules did not travel between repositories. Whether practices such as screenshot review, named benchmarks and rendered options work better is a question this single case raises but cannot settle.

Contributions

- A coded dataset of 242 design corrections across 15 products, with a stated inclusion rule, a screening record for all 250 candidates and the analysis scripts, published without private message text.
- A two-pole taxonomy of fifteen AI-default tells, from added excess (glow, status dots, decorative motion) to safe blandness (stock type, template sections, dead screens), grounded in the corrections.
- An observational measurement of written design rules against later corrections: most corrections came after a rule for that tell existed, rules rarely crossed repositories, and rates did not fall once a rule existed.
- A practitioner field guide of nine steering practices, each tied to the evidence behind it and to its limits.

Keywords. AI coding agents · generic design · design systems · design fixation · homogenisation · human-AI collaboration · visual design · experience report

1 Introduction

When one person builds many products by directing AI coding agents, the agents write most of the interface. The code usually works. The look often does not: it arrives in a recognisable house style that the person directing the work describes, again and again, as generic, basic, dead or "AI". This report studies that problem from the inside, using the records of one designer-developer, Yousof Selim (the author), who directed Claude Code and Codex across fifteen of his products between June and November 2026.

Controlled experiments give reason to expect such a house style. Writing with an instruction-tuned language model makes essays more alike [1]; AI-assisted short stories are individually better but collectively less diverse [2]; and people ideating with ChatGPT produce ideas that are less distinct from one another than people using a non-AI tool [3]. A practitioner evaluation of AI prototyping tools reports the visual counterpart: without detailed visual specification, the tools tend to produce designs that share a similar, generic look [4]. What is missing is evidence from sustained, real work: what the generic look consists of, what a designer does about it, and whether the usual remedy, writing rules into files the agents read, is followed by fewer corrections.

The study treats the author's own words and edits as data. It mines his typed messages from 87 surviving agent sessions, the persistent memory notes his agents wrote from his feedback, the design documents in eleven repositories and the commits that reverted designs he rejected. From these it codes 242 design corrections, each a rejection of one kind of pattern in one product on one date, and links each to the written rules that existed at the time.

Research questions

- **RQ1.** What does the generic "AI-default" look consist of, as seen in one designer's corrections across many products?
- **RQ2.** When he corrected a tell, did a written rule for it already exist, and were corrections of that tell less frequent once a rule existed?

- **RQ3.** How did he steer the agents, and how did the target of his corrections change as rules accumulated?

Summary of results

RQ1. Fifteen tells recurred, each in at least three products. The largest category was clutter and over-explaining copy (32 corrections), followed by dead, under-designed screens (26) and generic type (23); the familiar visual tells, such as glow (15) and status dots (9), were smaller. **RQ2.** For 173 of the 242 corrections (71%), a written rule covering that tell already existed: in 81 cases in the same product and in 92 only in another product. Across the 63 product and tell pairs that acquired a rule, he corrected the tell 0.35 times per review day up to the rule and 0.38 times after it; the comparison is observational and confounded, but it gives no sign that rules reduced corrections. **RQ3.** Screenshot or device review was the method seen most often (59 corrections). Type corrections rose from 4% of all corrections in June and July to 12% in October and 1 November, as earlier rules banned glow, gradients and decorative motion; the broader shift towards too-safe complaints was in the same direction but not statistically distinguishable from chance.

Contributions

- A coded dataset of 242 design corrections across 15 products, with a stated inclusion rule, a screening record and the scripts, published without private message text.
- A two-pole taxonomy of fifteen AI-default tells, from purple gradients and status dots to dead, under-designed screens, grounded in the corrections rather than in a public list.
- An observational measurement of written design rules against the corrections that followed them, showing that rules rarely travelled between repositories and were not followed by a lower correction rate within a product.
- A practitioner field guide of nine steering practices, each tied to the results that support it.

The study is a single-person, self-recorded case, and the AI assistant that coded it is from one of the two agent families studied. It cannot say how often agents produce generic design in general, and it cannot separate the agents' defaults from the author's moving taste. Sections 7 and 8 set out these limits.

---

2 Related work

2.1 Homogenisation of creative output

Three controlled studies frame the problem. Padmakumar and He had people write argumentative essays alone, with GPT-3 or with InstructGPT; essays co-written with the instruction-tuned model were less diverse, and the loss came mainly from the text the model contributed [1]. Doshi and Hauser found that access to AI-generated ideas made individual short stories more creative and better written while making them more similar to one another [2]. Anderson, Shah and Kreminski found that participants ideating with ChatGPT produced ideas that were less semantically distinct across users than those of participants using an alternative creativity support tool [3]. None of these studies concerns interfaces, and all are short laboratory tasks. The present study asks whether a similar pull is visible in interfaces built over months, and what a practitioner does once he notices it.

2.2 Design fixation

Jansson and Smith showed that designers given an example solution tended to reproduce its features, including its flaws, and named this design fixation [5]. Chen and colleagues argue that generative AI shows an analogous fixation that limits the novelty and diversity of its outputs [6]. In the present study the roles are unusual: the agent is the party anchored on familiar solutions, and the human reviewing its output is the one trying to break the anchor.

2.3 AI-assisted interface generation

Research on generating interface code has concentrated on whether the output compiles and renders. UICoder, for example, fine-tunes a language model with automated compiler and vision-model feedback to produce better SwiftUI code [7]. Visual distinctiveness is rarely the target. The Nielsen Norman Group evaluation cited above found that AI prototyping tools tend to converge on a generic look without detailed visual specification [4]. A 2026 preprint evaluated 120 interfaces from five generative interface tools and found that they converged in visual appearance and layout, with colour varying more [8]. These studies observe convergence in single generations. They do not follow a practitioner trying to steer out of it across many sessions and products, which is the gap this report addresses.

2.4 Why the look matters

Aesthetics shape perceived quality. Kurosu and Kashimura found that apparent usability was more strongly related to aesthetics than to inherent usability [9], and Tractinsky, Katz and Ikar found that the link between aesthetics and perceived usability held after use [10]. Lindgaard and colleagues showed that judgements of visual appeal form within about 50 milliseconds and are fairly stable [11]. Tuch and colleagues found that visual complexity

and prototypicality both shape first impressions, with simple, prototypical websites preferred [12], and Reinecke and Gajos showed that preferred colourfulness and complexity vary with country, age and education [13]. Two points follow. A generic look is judged quickly, so reviewing by screenshot is a reasonable proxy for how users first meet a product. And the preference for prototypical designs is in tension with the goal of looking unlike a template; the reversals in Section 5.7 show that tension in practice.

2.5 Design systems and agent instruction files as written rules

Design systems codify tokens, components and rules so that many contributors produce consistent work; Churchill describes them as a way to scale user experience across teams [14]. Coding agents now read their own instruction files. Chatlatanagulchai and colleagues analysed 253 `CLAUDE.md` files and found that they mostly hold operational commands, implementation notes and architecture [15], which suggests that design rules are rare in such files. In this study the equivalent is the set of design documents, memory notes and executable checks the author accumulated. The question is whether that codification is followed by what it is meant to achieve: fewer repeats of the same mistake.

2.6 Position of this report

Prior work establishes, in controlled settings, that generative models pull outputs towards a shared mode, and that design rules are the standard way to keep many contributors consistent. This report contributes field evidence that connects the two: a longitudinal, first-person record of what one designer rejected, which rules existed when he did, and how his target moved. It is an experience report with quantitative coding, not an experiment, and it should be read as hypothesis-generating.

---

3 Study context: one designer, two agent families, fifteen products

The author is a software engineering undergraduate who designs and builds his own products. Between June and November 2026 he directed AI coding agents, mainly Claude Code sessions and OpenAI Codex, across the products in Table 1. He wrote briefs and feedback, reviewed screenshots and device builds, and decided what shipped; the agents wrote most of the code and produced most first drafts of the interface. He often alternated agents on one repository and asked an inheriting agent to audit the previous agent's design before building on it.

Table 1. Products with at least one coded correction, and their state on 1 November 2026. Corrections are rows in the dataset.

Product	What it is	Corrections	State
Bupples	Group bill splitting (Flutter)	67	Shipped on iOS and Android; full design rework started 1 Nov
WayClub	Group management for clubs and communities (web, Flutter)	39	In development, pre-launch
Portfolio	His personal site	25	Live
Planket	Shared calendar for friends (web, Capacitor)	23	Web live; store builds pending
Partner portfolio	A portfolio site he directed for his partner	21	Live
OnPlan	Hangout planning (Flutter)	14	Paused in development
Adelante	Quotes and widgets (Flutter)	13	iOS build prepared; design rework started 1 Nov
Tajweed	Quran recitation learning (front end of a shared repository)	10	Redesign merged 13 Oct
NutriGo	Nutrition tracker (Flutter)	6	TestFlight builds
Experts Marine	Client website	5	Client project
Himalaya concept	Interview concept store	5	Demo deployed
Spell Quest	Children's spelling game (30-minute challenge)	5	Live
HACKFAM	Hackathon teammate matching	4	In development
Photoshoot	Photo booth (Electron, web)	3	Released 1.5.1

Product	What it is	Corrections	State
CareerOS	Evidence-backed career record	2	Live pilot

The distribution is uneven. Bupples was his longest-running product, and its surviving transcripts start in June; most other products began in late September or October, when the volume of corrections peaked (134 in October). Small counts for Photoshoot, CareerOS and HACKFAM reflect short design reviews in the surviving record; they say nothing about the quality of the agents' output there.

His agents kept a persistent memory: short notes, written by the agent after a session, that later sessions read on start. By 2 November there were 144 notes, 32 of them classed as feedback. Several repositories also carry written design rules: a `DESIGN.md` that opens by saying it exists so no agent "quietly regresses to AI defaults" (Bupples, 3 July), a pass/fail anti-slop manifesto with grep checks (WayClub, 3 August), and a 20,000-character design doctrine distilled from his feedback across projects (WayClub, 17 October). Some rules were executable: Bupples fails its test suite on an unlisted blur, a raw stock icon or a new literal animation duration, and HACKFAM's design system greps for uppercase transforms, positive letter-spacing and banned fonts.

4 Method

4.1 Sources

Four sources were read, all of them the author's own records. Two of them, the memory notes and many of the design documents, were written by agents from his feedback, so they are agent-written evidence about what he said, not independent evidence.

- **Session transcripts.** Claude Code keeps a JSONL transcript per session. A read-only miner (artifact `mine_transcripts.py`) read the 87 surviving top-level sessions, which run from 13 June to 2 November 2026. Earlier sessions had expired. The miner keeps only messages typed by the author: it skips subagent transcripts (whose "user" turns are written by another agent), tool results, task notifications, compaction summaries and harness markers. This left 2,194 of his messages. A design keyword pattern (generic, AI, basic, font, colour, motion, clutter, premium, glass, dot and similar; the exact expression is in the script) selected the design messages; after seven exact duplicates from resumed sessions were dropped, 537 remained. The session that commissioned this study was excluded. No assistant text or tool output was read for coding.

- **Memory notes.** The 144 notes in his agent memory, in particular the 32 feedback notes and the project notes for each product.
- **Design documents.** `DESIGN.md`, design system specifications, anti-slop documents, design rework specifications and hand-off notes in eleven repositories, with the date each was first committed.
- **Commits.** Commit messages matching a reversal pattern (remove, revert, drop, kill or replace near glow, glass, blur, sheen, font, icon, emoji, fade, entrance and similar), read to confirm what was reverted and when.

4.2 Screening

Among the 537 design messages, a second pattern selected candidate corrections: negative judgements (generic, basic, ugly, dull, dead, cringe, awkward, robotic, "don't like", "remove", "too much", cluttered, "over the top") in a first pass, and a broader list (out of place, janky, wacky, horrible, flat, plain, messy) in a second pass over the remainder. That gave 250 candidates, all read in full. Of these, 155 were coded, 9 were excluded because they were briefs written by another agent or pasted from another tool, and 86 were excluded because they described functional bugs rather than a design pattern. The screening outcome for every candidate is in the artifact.

4.3 Unit and inclusion rule

A correction is the author rejecting one kind of pattern in work an agent produced, or instructing an agent in advance to avoid it, in one product on one date. One message that rejects a font, clutter and a colour gives three corrections. A correction recorded in several places (his message, the memory note written from it, the commit that reverted it) is one row listing every source. Memory notes contribute a row only where no surviving message carries the same correction; this added 34 rows, mostly from WayClub and the partner portfolio, whose feedback often arrived as annotated screenshots or in sessions whose text did not survive. The final dataset has 242 rows: 202 with a transcript source, 79 with a memory source, 15 with a commit and 6 with a design document (rows can have several).

4.4 Coding scheme

Each row records the date, product, sources and evidence ids; a paraphrase of the pattern rejected and of the replacement; one tell category; the steering strategies visible in that correction; and whether the rejected pattern was one he had himself requested earlier. The tell taxonomy was built bottom-up from the first hundred corrections and then fixed; Table 2 lists it. Each tell is assigned to a pole. The too much pole holds things an

agent adds by default (colour, effects, dots, motion, gimmicks, loud surfaces). The too safe pole holds things an agent leaves generic by default (type, icons, marks, layouts, components, and screens that are simply dead).

Table 2. The tell taxonomy, with corrections and the number of products in which each tell was corrected.

Code	Tell	Pole	n	Products
CLU	Clutter, over-explaining and copy tells (helper sentences, em dashes, bare-noun buttons)	neither	32	10
DED	Dead or under-designed: restraint tipped into dull, plain, lifeless	too safe	26	11
TYP	Generic or robotic type: stock sans, monospace and tracked labels, tabular digits, thin or stretched	too safe	23	9
COL	Default colour: purple and AI blue, neon, gradients, warm paper, grey dark modes	too much	22	8
GEN	An unspecified "AI feel", no pattern named	neither	18	9
TPL	Template composition: same section pattern, card for everything, stat dashboards, wasted space	too safe	17	8
ICO	Generic icons: stock sets, emoji as icons, arrows for everything	too safe	17	11
MOT	Generic or decorative motion: roll-in jank, grow entrances, loops, ambient drift	too much	17	6
GLO	Decorative effects: glow, glass, blur, sheen, shading, textures	too much	15	4
SRF	Surface defaults: loud or nested outlines, mismatched shapes, flat boxes, see-through bars	too much	13	5
CMP	Stock components: default nav bar, numpad, spinner, dropdown, snackbar, app icon	too safe	11	3
DOT	Status dots, pills, badges and indicators	too much	9	5
IDN	Generic marks: typed logos, crude or borrowed marks	too safe	8	5
GIM	Gimmicks: effect catalogues, customisation sprawl, demo widgets	too much	8	3

Code	Tell	Pole	n	Products
NAV	Duplicate or misplaced entry points: two doors, two bars, stray buttons	neither	6	5

Strategies were coded only when the correction itself shows them: a named benchmark product (BEN), his own reference component or project as the grammar (REF), a list of named banned patterns (BAN), a research mandate (RES), rendered options or a prototype before building (PRO), screenshot or device review (SHOT), a full visual reset (RESET), and an audit of another agent's work (AUD). SHOT was also added automatically when a message carried an image or an uploaded screenshot path; 44 of the 59 SHOT codes come from this rule, so the SHOT count measures attachments as much as intent. Strategies are recorded only where visible, so a correction with no strategy code may still have used one.

4.5 Written rules and rule status

Twenty-seven written rules were dated: memory notes (dated by the session in which the rule was given), design documents (dated by first commit), and executable checks. Each rule lists the tells it covers and its scope, which is one product or, where he said so, all his products (two rules: the type and white rule of 17 October, which he applied to "his other products unless he says otherwise", and the no-dot rule of 19 October). Matching is by tell category, not by the exact pattern: a colour rule against purple counts as covering a later correction about warm paper. Rule status is therefore an upper bound on how often a correction repeated a specific written instruction. For each correction the analysis derives a rule status: in force if a rule covering that tell applied to that product before that date; elsewhere if such a rule existed only in another product (and so, unless he pointed to it, was not in the agent's context); none otherwise. For each product and tell pair with a rule, it counts corrections up to and including the rule date (the correction that produced the rule) and after it, and divides by the number of distinct days on which that product received any design correction, as a rough measure of review exposure.

4.6 Who coded, and a consistency check

The screening and coding were done by an AI assistant (Claude) from the author's records, under his direction; he did not code rows himself, and there is no second human coder. The coder belongs to one of the two agent families whose output is being judged, which is a potential conflict the reader should weigh. As a consistency check, an independent keyword coder, written before looking at the sample, assigned a tell to a seeded random sample of 60 rows from the pattern text alone. It agreed with the

assistant's codes on 48 of 60 rows (80%; Cohen's kappa 0.78). This shows that the codes are largely predictable from the paraphrased pattern. It is not a measure of inter-rater reliability: the paraphrases were written by the same coder, and the keyword list was written by the same assistant.

4.7 Statistics

Proportions are reported with 95% Wilson score intervals. The before-and-after comparison is summarised as a pooled rate ratio with an approximate 95% interval that assumes independent Poisson counts; corrections cluster within sessions, so the true uncertainty is larger and the interval is a rough guide only. Two sensitivity checks vary the comparison: restricting it to the 48 pairs with any review day after the rule, and excluding the same-day correction that produced the rule. Changes in proportions between periods are tested with two-sided Fisher exact tests. These tests are exploratory: the hypotheses were formed while reading the same data, and no correction for multiple comparisons is applied.

4.8 Replication

The artifact contains the three scripts. `mine_transcripts.py` applies the inclusion rule to a Claude Code transcript directory and writes a hashed public index and a private text file; `build_dataset.py` holds the coded rows and writes the data files; `analyse.py` recomputes every count, table and figure. Anyone can rerun the analysis from the published data. The coding step itself cannot be audited against the original text, which is not published; another researcher can replicate the procedure on their own transcripts, using the codebook and the inclusion rule, but not this exact coding.

---

5 Results

5.1 RQ1: What the generic look is made of

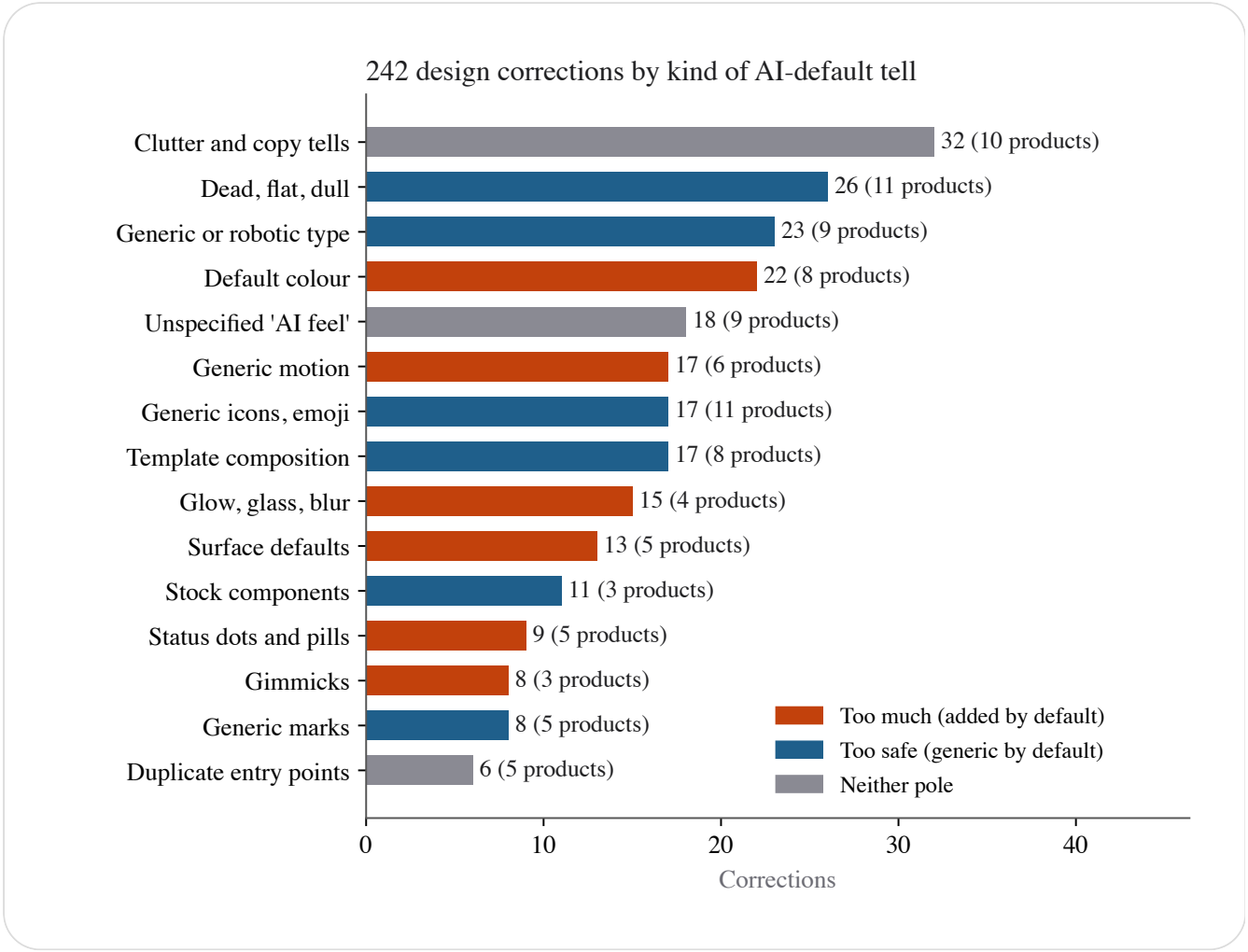


Figure 1. The 242 corrections by tell, coloured by pole, with the number of products in which each tell was corrected.

Figure 1 shows the distribution. The largest single category is not a visual effect at all: 32 corrections were about clutter and copy, such as a helper sentence under every heading, "too much text and overexplaining", a legend line explaining the colours, em dashes, and buttons labelled with a bare noun. Next come dead or under-designed screens (26) and generic or "robotic" type (23). The well-known visual tells are present but smaller: default colour 22, decorative glow, glass and blur 15, status dots and pills 9.

Counted by pole, 102 corrections (42%) were about things left too safe and 84 (35%) about things added in excess; 56 (23%) sit in neither pole. Only 18 corrections (7%) were unspecified "make it not feel AI" instructions: he usually named what was wrong, even when he did not yet say what he wanted instead, and 136 of the 242 rows show no steering strategy, only the rejection. The pole assignment is itself a coding judgement; for example, warm paper backgrounds sit under default colour, in the too-much pole, although they read as a safe choice.

The tells were not product-specific. Every one of the fifteen categories was corrected in at least three products, and dead screens and generic icons each in eleven (Figure 2). Status dots show a tell being generalised: he had asked for dots beside section headings and a location dot to come off his portfolio on 7 October and an availability dot off his partner's site on 15 October, then called the dot-and-label status chip "ai generic" in HACKFAM on 18 October, after which a rule against them was written for all his products (19 October).

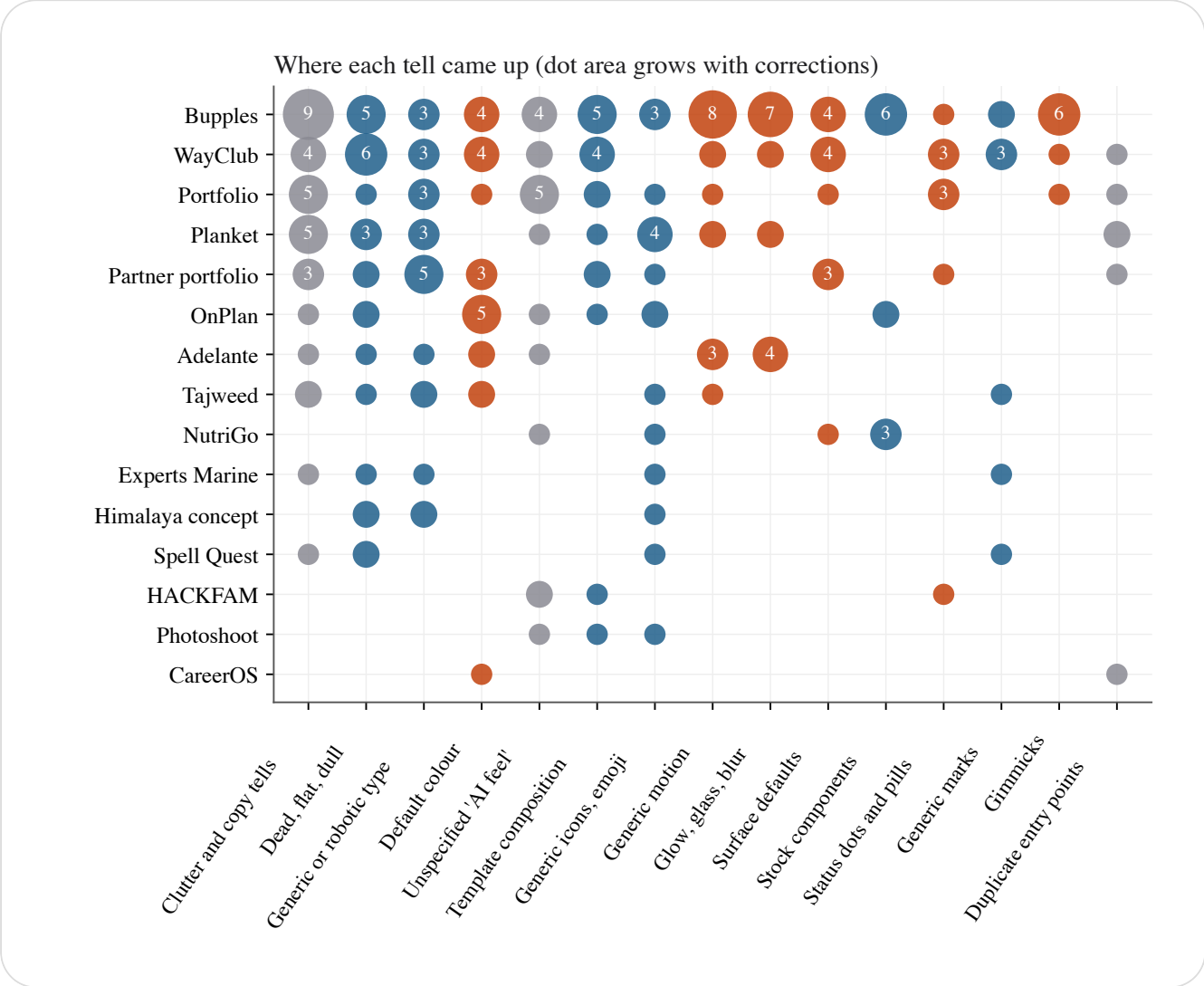


Figure 2. Where each tell came up, by product. Dot area grows with the number of corrections; numbers mark cells with three or more.

5.2 RQ2: Most corrections came after a rule for that tell existed

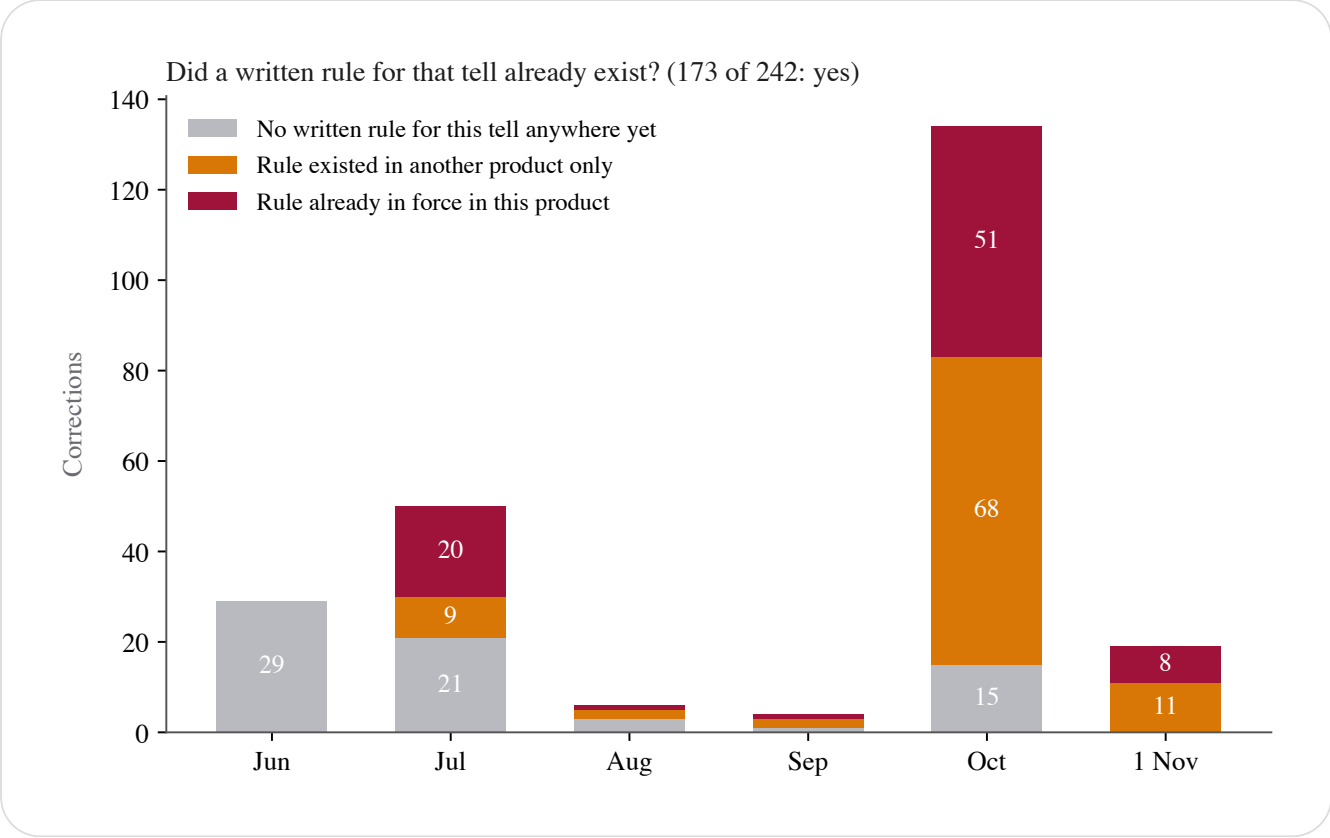


Figure 3. Rule status at the moment of each correction, by month. A rule counts if it covered the same tell category; August (6) and September (4) are too small to label.

Figure 3 answers the first half of RQ2. In June, before any written rule existed, none of the 29 corrections had a rule to point to. From July onwards, most corrections came after a rule for that tell had been written somewhere. Over the whole period, 81 corrections (33%; 95% CI 28% to 40%) were made while a rule for that tell was already in force in the same product, 92 (38%; 32% to 44%) while a rule existed only in another of his products, and 69 (29%) with no rule anywhere; together, 173 of 242 (71%; 66% to 77%). Because rules are matched by category (Section 4.5), these shares are upper bounds on exact repetition.

The "elsewhere" share is the most robust part of this result, since it does not depend on whether an agent read a rule: a rule in another repository was simply not in its context. Icon corrections are the extreme: 14 of 17 came after an icon rule existed in another product (Bupples' stock-icon ban of 3 July, OnPlan's no-emoji rule of 18 July, the portfolio's no-arrows rule of 14 October), but not in the product being reviewed. In effect, each new repository started from the agent's defaults. The cross-project doctrine written for WayClub on 17 October was explicitly reused for Planket and, on 1 November, for Adelante and Bupples; before that, rules travelled only when he asked an agent to read his other projects' research.

Table 3. Corrections by tell and rule status at the time of the correction (rules matched by tell category).

Tell	n	Rule in force here	Rule elsewhere only	No rule yet
Clutter and copy	32	11	18	3
Dead, flat, dull	26	5	12	9
Generic or robotic type	23	13	10	0
Default colour	22	10	9	3
Unspecified "AI feel"	18	1	6	11
Template composition	17	4	6	7
Generic icons	17	2	14	1
Generic motion	17	9	6	2
Glow, glass, blur	15	9	1	5
Surface defaults	13	8	2	3
Stock components	11	3	0	8
Status dots	9	3	3	3
Generic marks	8	3	0	5
Gimmicks	8	0	1	7
Duplicate entry points	6	0	4	2
All	**242**	**81**	**92**	**69**

5.3 RQ2: Corrections were no less frequent once a rule existed

The second half of RQ2 asks whether corrections became less frequent once a rule existed. Sixty-three product and tell pairs acquired a rule. Pooled over them, there were 87 corrections up to and including the rule date across 248 review days (0.35 per day) and 81 after it across 213 review days (0.38 per day), a rate ratio of 1.08 (approximate 95% CI 0.80 to 1.47). In neither sensitivity check was the rate after a rule lower. Restricted to the 48 pairs with any review day after the rule, the rates were 0.31 before and 0.38 after. Excluding the same-day correction that produced each rule, the rate before was 0.19

against 0.38 after. Of the 48 pairs with review days after their rule, 15 (31%) saw no further correction of that tell; 33 did.

This comparison cannot show that rules had no effect. Rules were written when a product was under heavy review, so the period after a rule contains the most intensive redesigns; a correction after a rule is often a subtler instance of the tell than the one that produced it; and review days are a crude measure of exposure. What it does show is that, in this record, writing a rule was not followed by fewer corrections of that tell.

Recurrence within a product was common. Of 102 product and tell pairs, 57 were corrected more than once, and those pairs hold 197 of the 242 corrections. Bupples' page-opening animation was rejected three times on 8 July before a transform-only transition replaced it. Planket's status-bar white fade was rejected twice on 25 October.

5.4 RQ2: Executable rules held their letter, not their intent

The strongest form of written rule was a test that fails the build. Bupples has three: a grep that forbids stock Material icons, an allowlist that fails on any unlisted backdrop blur, and a ratchet that fails on any new literal animation duration. After they landed on 3 July, Bupples still received 2 icon, 4 decorative-effect and 5 motion corrections. Three of the four effect corrections show how this happened. The blur allowlist checks one API, and the regressions arrived through others: a blur bar under a cover banner (26 July), a shade filter over profile photos (26 July) and a dot texture on solid backgrounds (21 July). An agent's project note from the same week records the workaround in plain terms: using a different filter for cover blur because it "stays outside the blur budget". The check held the letter of the rule; the rule was about the look, and the look regressed by another route. This is one documented case, not a rate, but it shows a mechanism by which an executable rule can be satisfied without its intent being met.

5.5 RQ3: How he steered

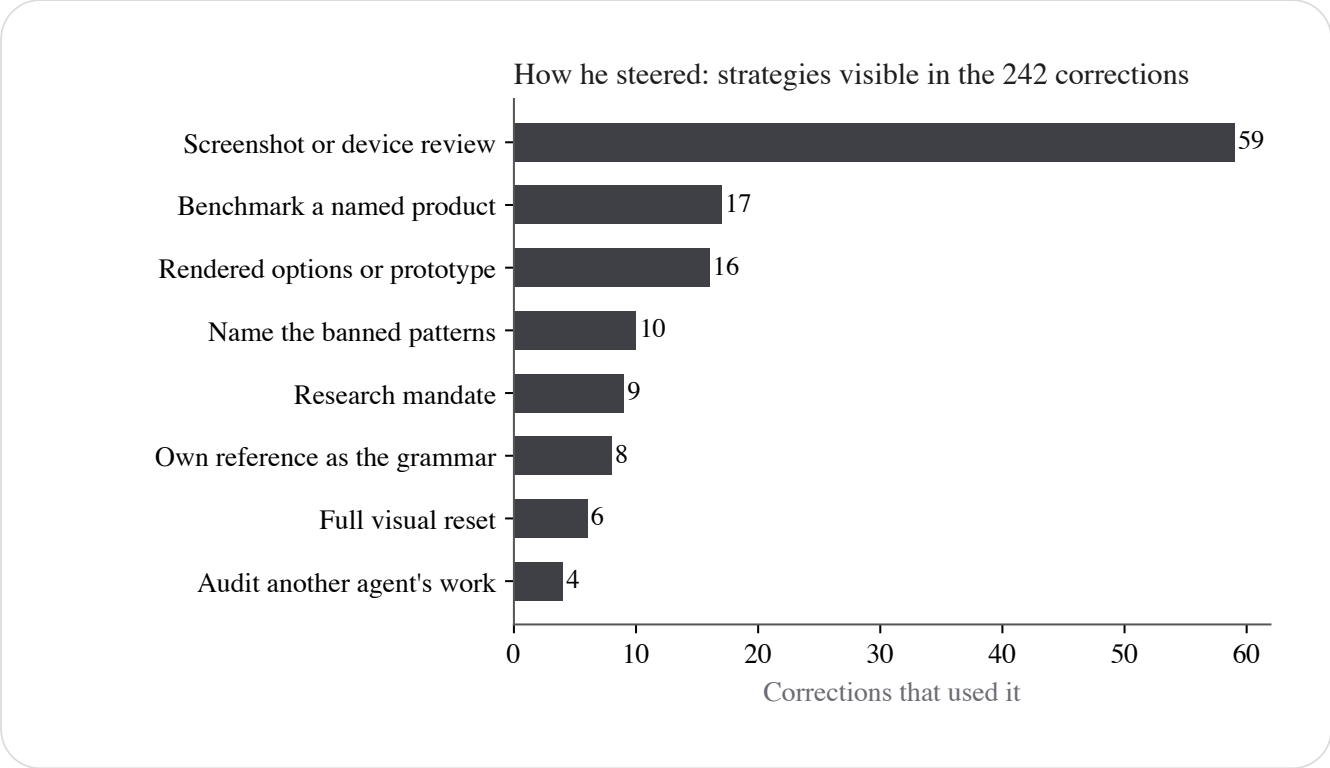


Figure 4. Steering strategies visible in the corrections. A correction can show several, and 136 show none; 44 of the 59 screenshot codes were added automatically from attached images.

Screenshot or device review was the most common visible method (59 corrections, 44 of them identified from an attached image). He judged by looking, often sending a crop with a short verdict: "this looks bad", "see how the lines look awkward". Named benchmarks came next (17): MacroFactor, Lose It and MyFitnessPal for NutriGo; Duolingo for Tajweed; Shopee and Amazon for the store; Luma and Meetup for WayClub; Apple's Photo Booth for Photoshoot. Rendered options and prototypes (16) became common from mid-October: a type specimen for Adelante, a shortlist of rounded faces for Planket, seven logo directions for WayClub, a rendered option sheet for his partner's site, a published prototype for the Bupples rework. Named banned lists (10) and research mandates (9) were fewer than their prominence in his memory notes suggests; most banned lists were written by agents from his corrections rather than typed by him. The data record which strategies he used, not whether they worked: no strategy was compared with another.

Two strategies are rare but distinctive. A reference component as grammar (8) appeared first on his portfolio, where he named the navigation pill, with its press, hover and focus behaviour, as the level of care he wanted everywhere; later he named whole projects as references, asking Planket for "the polish of tajweed" and Bupples for the motion work of Planket, HACKFAM and WayClub. Full visual resets (6) were reserved for products he judged beyond refinement: OnPlan in July, Adelante in September, WayClub on 17 October ("everything is changed") and Bupples on 1 November.

5.6 RQ3: The target moved towards blandness, especially type

Table 4. Corrections by pole and period. Type is a subset of the too-safe pole; the remaining corrections in each period sit in neither pole.

Period	Corrections	Too much	Too safe	Of which type
June to July	79	34	32	3
August to September	10	3	3	1
October to 1 November	153	47	67	19

In June and July the two poles were close to level: 32 too-safe against 34 too-much corrections, a too-safe share of 48% (95% CI 37% to 60%). In October and 1 November the too-safe pole led by 67 to 47, a share of 59% (50% to 67%). This difference is in the expected direction but could be chance (Fisher exact test, p = 0.21). The change in type is clearer: type corrections were 3 of 79 corrections (4%) in June and July and 19 of 153 (12%) in October and 1 November (p = 0.03, exploratory). Raw counts should not be compared directly, because October alone holds 134 corrections. The records suggest a displacement mechanism, though they cannot establish it: once the obvious tells were banned, agents reached for the next most common choice. After the WayClub redesign brief had named Inter, Manrope and Space Grotesk as tells, the design panel's agents reached for IBM Plex Mono and Bricolage Grotesque, and three of the four design theses prescribed warm paper backgrounds, so he had to ban those too. The Himalaya concept used Bricolage Grotesque and he called it generic and "used alot by ai" (15 October). The WayClub doctrine ended up with an explicit category for these: "second-order taste", the defaults that replace the first-order defaults once those are banned. On type, the agents moved from Inter to the fashionable alternatives to Inter.

The same pattern may account for much of the "dead" category. Restraint is the standard advice against AI excess, and the agents applied it. The Himalaya store came back with a white label-stock canvas, one accent and no motion, and he rejected it as dead, saying it did not look or act like Shopee or Amazon (14 October). A research-compliant banner for his partner's portfolio, plain type on a flat ground, was "generic and bad super bad" (15 October). WayClub, after its motion was rationed to three moments, was "not energized and dull" compared with Luma (20 October). In these cases the rejected default was not decoration but its absence.

5.7 RQ3: His own taste moved

Table 5. Reversals: a pattern he asked for, or a ban he agreed, that he later reversed.

Product	Earlier	Later
Tajweed	Asked for a warm off-white light mode (20 Jun)	Rejected warm paper as generic and dull (13 Oct)
Bupples	Asked for a zoom-out entrance on bubbles (29 Jun)	Entrance removed after five failed fixes (30 Jun)
Bupples	Monospace money digits in the rebuild concept (29 Jun)	Called robotic; rounded digits default (18 Jul)
Bupples	Asked for native Liquid Glass on the bar and every button (18 Jul)	Mode removed entirely (1 Nov)
Adelante	Asked for native Liquid Glass (18 Jul) and a frosted glow identity (30 Sep)	Frosted cards called "cringe looking AI frost" (1 Nov)
WayClub	No glass or blur on content (17 Oct)	Frost the way Luma and Meetup use it (25 Oct)
WayClub	Motion rationed to three signature moments (19 Oct)	Many custom mark animations wanted (20 Oct)
Himalaya	Anti-slop restraint applied (14 Oct)	Rejected as dead (14 Oct)
Partner portfolio	Subtraction: no monospace, no motifs (15 Oct)	Chose dot-matrix and monospace labels from rendered options (15 Oct)

Not every correction was against the agent. Two coded rows reject a pattern he had himself requested, and Table 5 lists nine reversals in total, four of them about glass and glow. Glass is the clearest case of a pattern that is a tell in one form and a requirement in another. He banned glow in Bupples in June, asked for Apple's native Liquid Glass in July, removed it in November; WayClub banned glass on content on 17 October and adopted Luma's tinted frost eight days later. His rule was never "no glass". It was closer to "no glass that could be anyone's": frost tinted by the page's own poster colour was wanted, a generic frosted slab with a hairline under it was not. The user profile his agents kept in August recorded a liking for glow, glass and magnetic hovers; by October several of those were on his banned lists. The taxonomy therefore captures what he rejected at the time, not a fixed list of forbidden patterns.

5.8 Three cases in detail

The bubble that would not grow (Bupples, June). On 29 June he asked for the people bubbles on the session screen to have an opening animation, zooming out to their normal size. The agent added a scale-from-zero entrance. It completed only for bubbles with a profile photo, because late image loads triggered the rebuilds that finished the animation; bubbles for offline guests stayed small. Over 30 June five commits tried to rescue the entrance (a dedicated animation controller, a timer guard, raised size floors) before a sixth removed it. The memory note written afterwards forbids reintroducing any grow or fade keyed to spawn time. This case is one of the two coded rows where the rejected pattern was his own request. In this case decorative motion failed on substance as well as taste: an entrance that carried no meaning was also a source of state bugs.

"Everything is changed" (WayClub, October). By 17 October WayClub had a violet-black dark panel, a "Hello, Alex" greeting dashboard with a stats row, placeholder brackets for posters and reassuring microcopy. He asked that it not have "the ai feeling" and that everything be changed. Within the same hour he sent a crop of a tracked uppercase monospace label on a cream ground and banned that type, generic fonts and warm white, then added that it needed no clutter or over-explaining. The agent wrote these into memory and into a cross-project doctrine distilled from his feedback on seven other products. The redesign then ran for two weeks and produced 29 further WayClub corrections, split evenly between the poles (13 each). The largest single group was dead or dull surfaces (5): a dull header, a dead organiser section, a flat logo, backgrounds that felt dead. The reset removed the first-order tells; it did not supply the energy he wanted. In the record, the corrections that followed name Luma and Meetup as benchmarks and point to a prototype he could try on a phone.

Liquid Glass (Bupples, July to November). On 18 July, when Apple's Liquid Glass shipped, he asked for the native glass navigation bar and then for glass on every button. The agents built a native tab bar, a glass surface token and an appearance switch, defaulting to off at his request. On 1 November, starting a full design rework, he asked for the mode to be removed entirely, about forty files and two native plugins. Nothing in the record says the glass was badly built. Between July and November his products had accumulated rules against glass on content and glow, and the doctrine of 17 October allowed glass only on floating chrome. Glass as an app-wide mode had become, by his own later standard, the kind of default he was steering away from.

6 Discussion

6.1 What the answers mean

The three results fit together. The generic look is not one thing to ban but two opposite failure modes (RQ1). Written rules were already in place for most of the tells he corrected, and corrections did not become less frequent once a rule existed (RQ2). And as rules removed the obvious excesses, the corrections that remained were more often about type and blandness (RQ3). One reading, consistent with the records but not proven by them, is that rules move an agent away from a named default without telling it what to do instead, so it lands on the next most common choice.

The strongest single implication concerns scope. Rules were written per repository and did not travel, so 38% of corrections concerned a tell already ruled against elsewhere. For the people who build agent tooling, this argues for design rules that can be inherited across a person's or a team's repositories, in the same way that coding conventions are, rather than rewritten per project. The case of the blur allowlist (Section 5.4) argues, more tentatively, for checks against rendered output as well as code patterns.

6.2 A field guide for practitioners

The following practices are drawn from this record. Each is tied to the results that support it, and each is limited by the single-person, observational evidence behind it; none was tested against an alternative.

1. **Expect both failure modes, not one.** Agents fail by adding (glow, gradients, dots, fade-ins, effect catalogues) and by playing safe (stock type, stock icons, template sections, dead screens). Here the second outnumbered the first, 102 to 84, and type complaints grew as the first was banned (Section 5.6). A brief that only says "no AI slop" invites the safe failure. Say what the product should have as well as what it must not.
2. **Ban by name, then expect the next default.** Named bans work on the named item and push the agent to the nearest unbanned alternative. When you ban Inter, check for Bricolage and Instrument Serif; when you ban purple, check for cream paper and terracotta (Section 5.6). Keep a "second-order" list.
3. **Carry the rules into every new repository on day one.** 38% of corrections (92 of 242) concerned a tell that already had a rule in another product's repository but not in the one under review (Section 5.2). A cross-project doctrine, read by every agent at the start of every new product, avoids rediscovering the same tells in every repository. This record cannot show that such a doctrine reduces corrections; it was adopted too late to test.
4. **Do not count on a written rule to stop a tell.** Correction rates were 0.35 per review day before a rule and 0.38 after (Section 5.3). Treat the rule as a record of taste for the next agent, and keep reviewing.
5. **Write checks against the look, not the API.** In the one documented case, an executable rule held its letter and the look regressed by another route (Section 5.4). Where a rule can

be checked, check the rendered result (screenshot review against the rule) as well as the code pattern.

6. **Judge by screenshot, and send the screenshot.** Screenshot and device review was the most used method (59 corrections). First impressions of visual appeal form in about 50 milliseconds [11]; reviewing renders rather than diffs matches how users first judge a product.
7. **Name a product to beat, and a component of your own to match.** Named benchmarks (17) and reference components (8) gave agents a concrete target, such as Luma's tinted pages or his own navigation pill's press and focus behaviour. A benchmark sets the ambition; a reference component sets the grammar.
8. **Ask for rendered options before a direction is built.** Several fonts, logos and identities were settled by choosing from a rendered sheet (Adelante's face, Planket's face, WayClub's mark). Choosing from rendered options surfaces the agent's defaults before they spread across every screen; this record does not measure whether it is faster.
9. **Write the "why" next to the ban, because your taste will move.** Nine reversals, four of them about glass, show that a pattern can be a tell in one form and the right choice in another (Section 5.7). A rule that records the reason ("no frost that could be anyone's; frost tinted from the page's own colour is fine") survives a change of mind better than a bare ban.

---

7 Threats to validity

Construct validity. "AI default" is inferred, not observed. The study codes what the author rejected in agent-produced work; it does not test whether an agent produces a pattern more often than a human designer would. Some rejected patterns (a misplaced button, a crowded settle-up screen) are ordinary design problems that any first draft might have. The taxonomy groups them with recognisable AI tells because he did, but a reader should not take every row as evidence about model behaviour. Rule status matches rules to corrections by tell category, so "a rule existed" means a rule about that kind of tell, not necessarily about that exact pattern.

Internal validity. The before-and-after comparison is observational and confounded. Rules were written when a product was under heavy review, so the period after a rule includes the most intensive redesigns, and later corrections are often finer-grained; a rule's date marks a burst of attention as much as a change in the agent's instructions. Review days, counted only on days with a correction, are a weak and endogenous proxy

for exposure. The rule-status measure also depends on dating memory notes by the session that produced them; a few notes were edited later, and the record does not show whether an agent actually read a given rule in a given session. Nothing in this study supports a causal claim in either direction.

Coding and dependence on agent-written evidence. One AI assistant screened, paraphrased and coded every row, and it belongs to one of the two agent families whose output is judged. The keyword consistency check (kappa 0.78 on 60 rows) shows that the codes are predictable from the paraphrases, but the paraphrases came from the same coder, so this is a check of internal consistency, not of agreement between independent judges. The 34 memory-only rows, and the rule dates taken from memory notes, rest on text that agents wrote about his feedback and may tidy or overstate it. Splitting a message into one row per tell is a judgement; a different splitting rule would change counts by category.

Conclusion validity. The confidence intervals and tests in Section 5 assume independent observations, which corrections clustered within sessions and products are not, so they understate uncertainty. The tests were chosen after reading the data and are uncorrected for multiple comparisons; the type result ($p = 0.03$) in particular should be treated as a lead, not a finding.

Sources and selection. Transcripts before 13 June did not survive. The keyword filters will have missed corrections phrased without any listed word; the second pass found 10 more coded messages among 47 candidates, which suggests the first pass missed some. Corrections made by editing code directly are absent.

External validity. Everything here comes from one person, two agent families and fifteen products in five months. The products are consumer apps and personal sites built by the same person, so the tells will reflect his genre and taste as much as the agents' defaults. Agent models changed during the period, and the results may not hold for later versions.

---

8 Limitations

This is a study of one person (n = 1), who is also the author. It cannot estimate how common any tell is in AI output, and it cannot compare his practice with anyone else's.

The records are self-selected. They exist because he kept feedback in agent memory and because some transcripts happened to survive. Corrections made in sessions that expired, or by editing code himself, are absent.

Taste is subjective. Several rows are about preference rather than quality (a font he disliked, an accent colour), and the reversals in Table 5 show that his preferences moved. The study treats his rejections as data about how he steered, not as ground truth about good design. Preferences for colour and complexity vary with country, age and education [13], so his rules are not universal rules either.

The written evidence is largely agent-produced, and so is the analysis: the coding, the scripts and the first draft of this report were produced by an AI assistant under his direction. No independent human checked the coding.

There are no outcome measures. The study does not test whether the corrected designs were better for users, only that he rejected the earlier ones, and it does not test whether any steering strategy works better than another. Future work could pair a correction dataset like this with user ratings of before and after screens, have a second human code a sample from the original text, match rules to corrections at the level of the exact pattern, or run the same coding on several designers to see which tells are shared.

---

9 Conclusion

Across fifteen products and 242 corrections, one designer repeatedly pushed AI agents away from their defaults, and the same kinds of default kept returning. For 71% of corrections a rule for that kind of tell had already been written, but often in another repository, and corrections were no less frequent within a product once a rule existed. The record cannot say why; the timing of rules confounds any comparison. What did change was the kind of default he corrected: as the obvious excesses were banned, type and over-restrained, dead screens took a larger share. For practitioners, the case suggests treating written rules as a record of taste rather than a guarantee, carrying them into every new repository, and keeping visual review in the loop. Whether these practices reduce corrections is a question for studies with more than one designer and a design that can test it.

Artifact

The artifact at /research/beyond-the-default/artifact/ contains the 242 coded corrections, the 27 dated rules, the nine reversals, the codebook, a hashed index of the 537 selected messages, the screening outcome for all 250 candidates, the consistency-check sample, and the three scripts (mining, dataset and analysis) that regenerate every number and figure in this report. Raw message text is not published because it contains other people's names, private links and unreleased work; every row instead carries evidence ids, a paraphrase and, for 20 rows, a short quotation of the author's own words.

References

1. V. Padmakumar and H. He. Does writing with language models reduce content diversity? In The Twelfth International Conference on Learning Representations (ICLR), 2024. proceedings.iclr.cc/paper_files/paper/2024/hash/02dec8877fb7c6aa9a79f81661baca7c-Abstract-Conference.html
2. A. R. Doshi and O. P. Hauser. Generative AI enhances individual creativity but reduces the collective diversity of novel content. Science Advances 10(28), eadn5290, 2024. doi.org/10.1126/sciadv.adn5290
3. B. R. Anderson, J. H. Shah and M. Kreminski. Homogenization effects of large language models on human creative ideation. In Proceedings of the 16th Conference on Creativity & Cognition (C&C '24), pp. 413–425, 2024. doi.org/10.1145/3635636.3656204
4. H.-H. Wang and M. Brown. Good from afar, but far from good: AI prototyping in real design contexts. Nielsen Norman Group, 24 October 2025. nngroup.com/articles/ai-prototyping
5. D. G. Jansson and S. M. Smith. Design fixation. Design Studies 12(1), pp. 3–11, 1991. doi.org/10.1016/0142-694X(91)90003-F
6. L. Chen, Y. Song, C. Zheng, Q. Jing, P. Hansen and L. Sun. Understanding design fixation in generative AI. arXiv:2502.05870, 2025 (preprint). arxiv.org/abs/2502.05870
7. J. Wu, E. Schoop, A. Leung, T. Barik, J. P. Bigham and J. Nichols. UICoder: Finetuning large language models to generate user interface code through automated feedback. In

Proceedings of NAACL 2024 (Long Papers), pp. 7511–7525, 2024. doi.org/10.18653/v1/2024.naacl-long.417

8. K. Imteyaz, K. Imteyaz, N. Rajpal, K. Shaikh, M. Muller and S. Savage. Design theater: Evaluating the gap between user-facing design reasoning and implementation in generative UI tools. arXiv:2607.22928, 2026 (preprint). arxiv.org/abs/2607.22928
9. M. Kurosu and K. Kashimura. Apparent usability vs. inherent usability: Experimental analysis on the determinants of the apparent usability. In CHI '95 Conference Companion, pp. 292–293, 1995. doi.org/10.1145/223355.223680
10. N. Tractinsky, A. S. Katz and D. Ikar. What is beautiful is usable. Interacting with Computers 13(2), pp. 127–145, 2000. doi.org/10.1016/S0953-5438(00)00031-X
11. G. Lindgaard, G. Fernandes, C. Dudek and J. Brown. Attention web designers: You have 50 milliseconds to make a good first impression! Behaviour & Information Technology 25(2), pp. 115–126, 2006. doi.org/10.1080/01449290500330448
12. A. N. Tuch, E. E. Presslaber, M. Stöcklin, K. Opwis and J. A. Bargas-Avila. The role of visual complexity and prototypicality regarding first impression of websites: Working towards understanding aesthetic judgments. International Journal of Human-Computer Studies 70(11), pp. 794–811, 2012. doi.org/10.1016/j.ijhcs.2012.06.003
13. K. Reinecke and K. Z. Gajos. Quantifying visual preferences around the world. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14), pp. 11–20, 2014. doi.org/10.1145/2556288.2557052
14. E. F. Churchill. Scaling UX with design systems. Interactions 26(5), pp. 22–23, 2019. doi.org/10.1145/3352681
15. W. Chatlatanagulchai, K. Thonglek, B. Reid, Y. Kashiwa, P. Leelaprute, A. Rungsawang, B. Manaskasemsak and H. Iida. On the use of agentic coding manifests: An empirical study of Claude Code. In Product-Focused Software Process Improvement (PROFES 2025), Lecture Notes in Computer Science, pp. 543–551, Springer, 2025. doi.org/10.1007/978-3-032-12089-2_40

Cite and reuse

Authorship and AI assistance. Yousof Selim is the designer whose feedback is studied, and he directed this study. The products studied were built largely by AI coding agents (Claude Code, Codex) under his direction. The screening, coding, analysis scripts and the draft of this report were

produced with an AI assistant (Claude, from the same family as one of the agents studied) under his direction and review. No second human coder was involved, and the agent memory notes used as evidence were written by agents. This is a technical report and has not been peer reviewed.

```
@techreport{selim2026beyond,
  author      = {Yousof Selim},
  title       = {Beyond the Default: A Coded Study of 242 Design Correct
  institution = {Sunway University (Lancaster University dual degree), M
  type        = {Technical report},
  number      = {v1.0},
  year        = {2026},
  url         = {https://yousofselim.com/research/beyond-the-default/}
}
```