Studio DNA

https://gerolds.github.io/textbook/textbook/posts/studio-dna/

Contents

Studio DNA

1. The observation

Studios develop institutional DNA. Over years of solving the same class of problems, a team accumulates habits of taste, production reflexes, hiring patterns, technical infrastructure, risk tolerance, and assumptions about what players will put up with. That DNA is not a label. It describes what the team has learned to do, and by implication, what it hasn't needed to learn.

This essay is a way to read that DNA. Not a taxonomy, but a set of recognizable patterns that make it easier to ask: *what is this team adapted for, and what happens when the next project demands something different?*

---

2. What the DNA describes

The common ways of sorting studios — AAA versus indie, big versus small — are market descriptions. They tell you about budget and headcount, not about what problems the team is organized to solve.

A better map would plot studios by **what kinds of questions they naturally ask.** Not what genre they ship in, but what design problems they've trained themselves to notice, and which ones they've trained themselves to ignore.

A team that spent a decade making systems-heavy strategy games has developed instincts for interlocking mechanics, emergent behavior, and economy balance. Those instincts are real, but specific. The same team may have little practice with moment-to-moment feel, emotional pacing, or the craft of making a single action satisfying in the hands. Not because the people can't learn, but because the organization never needed to — and so never built the reflexes, the hiring pipeline, the review culture, or the vocabulary.

A studio's past success describes local fitness within a specific design ecosystem, not universal capability across

all games.

This matters because success in one game is often read as evidence of general competence — “being good at games.” When that assumption meets a genre shift, the result is a team whose instincts are well-developed but pointed at the wrong problem. Not a failure of talent. A mismatch between adaptation and environment.

---

3. The DNA families

Not a definitive list. A set of recognizable clusters — families of institutional adaptation that show up repeatedly. Most studios aren’t pure examples of any one family; they live between two or three. The families are useful because they make visible what a team is optimized for and what its blind spots are likely to be.

3.1. Systems Architects

Teams that excel at interlocking mechanics, simulation logic, economies, progression balance, AI interactions, and emergent play. Their games live or die on whether the systems keep producing interesting situations without hand-authoring.

Strategy games, management sims, colony sims, immersive sims, sandbox survivals, grand strategy, some RPGs.

The question they naturally ask: *What happens when these rules collide?*

Strength: depth. They build machines that generate stories. Blind spot: immediacy. Moment-to-moment feel can be dry. Onboarding can be hostile. Presentation lags behind mechanical sophistication. They’ll build an elegant food-distribution model and forget that the player can’t tell why their villagers are starving.

3.2. Feelsmiths

Artisans of input response: timing, animation, impact, control latency, readability, mechanical juice. They care about whether

moving, jumping, shooting, dodging, or striking feels right in the hands.

Platformers, action games, character action, fighting games, high-polish shooters, racing games, arcade-inspired design.

The question they naturally ask: *Does this action feel right?*

Strength: embodiment. They make verbs feel good. Blind spot: larger structural design. A studio can ship fantastic combat or movement and still struggle with pacing, economy, encounter variety, or long-form progression. They can make you love pressing buttons before they've figured out why you'd still be pressing them ten hours later.

3.3. **Toymakers**

Skilled at playful interaction, object affordance, experimentation, physics comedy, tactile delight, and open-ended messing around. Their games are fun to touch before they're fun to master.

Toy-like sims, VR interaction games, sandbox puzzlers, social chaos games, physics games, party titles.

The question they naturally ask: *What happens if the player just starts messing with everything?*

Strength: curiosity. They make players want to poke the world with a stick. Blind spot: durability. Novelty carries a lot, but the spell fades once the toy has been understood. Good at building a machine that bonks and squeaks, less certain to build a structure that holds attention for twenty hours.

3.4. **Dramatists**

Teams that think in scenes, pacing, emotional beats, character arcs, worldbuilding, tonal control, and authored revelation. They know how to make a player care what happens next.

Narrative adventures, cinematic action games, RPGs with strong authored storytelling, visual novels, horror games.

The question they naturally ask: *What is the player feeling in this scene?*

Strength: emotional momentum. Blind spot: systemic freedom. Pushed into emergent or player-authored spaces, they over-script, overconstrain, or panic when players start coloring outside the lines. They build cathedrals, not weather systems.

3.5. **Social Engineers**

Optimized around player-to-player dynamics: matchmaking, cooperation, competition, status, retention loops, spectator value, virality, community incentives.

Multiplayer shooters, MOBAs, MMOs, extraction games, party games, sports games, live social sandboxes.

The question they naturally ask: *What do players do to each other?*

Strength: understanding that the “game” is partly the other humans. Blind spot: single-player content may feel thin, because the design instinct assumes social energy will fill gaps.

3.6. **Service Operators**

This family overlaps with others. Its defining trait is operational: cadence, seasonal content, monetization design, retention analysis, economy maintenance, community management, and balancing a game as a living system over years.

Live-service games, long-tail online games, card games, gachas, persistent multiplayer ecosystems.

The question they naturally ask: *What keeps this ecosystem alive?*

Strength: adaptation. Blind spot: the feedback loop can narrow to metrics. When every decision is tested against retention and conversion data, design drifts toward what’s measurable rather than what’s meaningful. The game stays alive, but the reasons to play slowly flatten.

3.7. **World Simulators**

Distinct from Systems Architects because the central value is plausibility and coherent world-modeling rather than abstract strategic elegance. They care about “how this thing really works” — whether the thing is farming, trucking, flying, urban planning, ballistics, or historical process.

Hardcore sims, serious management games, historical modeling, sim-lite worlds, authenticity-first experiences.

The question they naturally ask: *Does this model earn belief?*

Strength: trust. Players believe in the world. Blind spot: realism can become an alibi for friction, opacity, and bad pacing.

3.8. **Experience Sculptors**

Design around mood, atmosphere, aesthetic cohesion, discovery, and sensory texture. Less interested in challenge or long-term systems than in constructing a state of mind.

Walking sims, exploratory adventures, atmospheric horror, art games, meditative experiences, some puzzle adventures.

The question they naturally ask: *What state of mind are we creating?*

Strength: resonance. They can make thirty minutes feel unforgettable. Blind spot: replayability or systemic robustness. They’re not building a machine; they’re composing a weather pattern.

3.9. **Puzzlesmiths**

Specialized around clarity, rule-space, constraint, revelation, and the slow magic trick of making a player understand something elegant.

Logic puzzles, programming games, deduction games, sokoban-likes, systemic puzzlers, escape-room structures.

The question they naturally ask: *What understanding am I trying to unlock?*

Strength: intellectual architecture. Blind spot: breadth of appeal or production flexibility. Puzzle design is specific enough that a strong puzzle studio would be poorly equipped to make a compelling brawler. Different circus, different clowns.

3.10. **Spectacle Builders**

Excellent at scale, scripted sequences, visual payoff, destruction, orchestration, and making the player feel carried through a roller coaster of high-energy moments.

Blockbuster action, set-piece shooters, action-adventure, some open-world event design.

The question they naturally ask: *Is this moment landing?*

Strength: momentum. Blind spot: outside authored peaks, the valleys may be undercooked. A team can get so good at fireworks that it forgets campfires matter too.

3.11. **Content Industrialists**

Optimized for large asset pipelines, massive amounts of authored content, broad feature coverage, and enormous production coordination. Open-world games need this trait more than people realize.

Giant RPGs, open-world adventures, loot-heavy games, licensed content games, massive content libraries.

The question they naturally ask: *Can we fill this world?*

Strength: scale. Blind spot: uneven density and formula drift. They know how to build continents, but not always how to make every square meter necessary.

---

4. **The poles**

A spatial intuition for how the families relate. Imagine a few poles with families clustered around them:

- **Authored experience:** Dramatists, Spectacle Builders, Experience Sculptors. The work is shaped, sequenced, controlled.
- **Mechanical mastery:** Feelsmiths, Puzzlesmiths, some racing and fighting specialists. The work is about how actions feel and resolve.
- **Systemic emergence:** Systems Architects, World Simulators, sandbox builders. The work is about what the rules produce.
- **Human networks:** Social Engineers, Service Operators. The work is about what happens between players.

Toymakers and Content Industrialists orbit between these poles, solving different but widely reusable problems.

Most studios live between two or three poles. Very few sit comfortably near all of them. The studios that do are the ones people mythologize — but even those teams have usually solved one specific organizational problem: getting different DNA families to coexist in tension without strangling each other.

---

5. **Why genre shifts are hard**

This framing makes a prediction: genre shifts should be among the hardest things a studio can attempt, and the difficulty should be predictable from the distance between the DNA families involved.

It holds. A studio raised on authored spectacle struggles with emergent systems because it keeps trying to script what should be allowed to happen. A systems-heavy team struggles with onboarding because it assumes the player will invest effort before the game has earned it. A toy-driven team delivers wonderful first impressions and weak long-term structure. A narrative team resists player freedom because it has learned that authored control is what makes scenes land.

These aren’t failures of talent. They’re failures of environment. The team’s instincts — honed over years and embedded in hiring practices, review culture, tool choices, and implicit definitions of “good” — are adapted to the wrong problem.

The tell is invisible before the shift and obvious after. The team doesn’t suddenly become worse. It becomes worse *at the specific thing the new genre demands*, while remaining excellent at something nobody asked for. The combat feels great but the economy is incoherent. The systems are deep but the first hour is impenetrable. The world is atmospheric but there’s nothing to do in it. The content is vast but nothing surprises.

Each failure has a DNA signature. Once you learn to read them, the failures stop looking like bad luck and start looking like organizational physics.

---

6. **Genres as organizational psychology**

A further implication: game genres are not just market categories. They’re expressions of organizational psychology.

A tight fighting game comes from a culture that values precision and matchup integrity. A live-service MMO comes from a culture comfortable with operations, metrics, and player sociology. A narrative adventure comes from people who think in scenes and emotional cadence. A colony sim comes from people who trust systems to create stories.

The genre doesn’t just describe the product. It describes the kind of team that can make the product well. “Well” here doesn’t mean competently. It means with the instincts, vocabulary, review habits, and failure-detection reflexes that the genre requires.

Two studios can have identical budgets, identical team sizes, and identical stated ambitions, and produce wildly different results. The budget and the ambition are visible. The DNA is not. But the DNA determines whether the team naturally asks the right questions or has to be reminded to ask them — which is never quite the same thing.

---

7. **How to use this**

The point isn’t to label yourself “We are Feelsmiths” and file it away. Labels close inquiry; the point is to open it.

Notice which questions your team asks *without being prompted*, and which it has to be reminded to ask. The first set is your DNA. The second is your blind spot.

A few diagnostics:

When a prototype isn't working, what does the team reach for first? If the instinct is "make the controls tighter," you're near the Feelsmiths pole. If it's "add more systems depth," the Systems Architects. If it's "write a better narrative wrapper," the Dramatists. The instinct reveals the family, because it's what the team *trusts* to fix things.

What does the team notice in other people's games? When someone plays a competitor's game and comes back excited, what are they excited about? The feel? The systems? The story? The social dynamics? The atmosphere? That excitement reads the DNA.

What feedback does the team resist? If playtesters say the game feels stiff and the team says "that's not the point," the DNA probably isn't Feelsmiths. If playtesters say the systems are shallow and the team says "we're not making a spreadsheet game," the DNA probably isn't Systems Architects. The feedback you resist most fluently is the feedback your DNA is least equipped to process.

What does the team's "good enough" look like? Every team has a threshold below which work isn't acceptable. What triggers it? If janky animation makes everyone wince, feel is in the DNA. If an incoherent economy makes everyone wince, systems depth is. If a flat emotional beat makes everyone wince, drama is. What the team refuses to ship reveals what the team actually values.

None of this tells you what to do. It tells you what you're likely to be bad at without knowing it — and that's worth having, because the failures that kill projects are rarely the ones the team can see coming. They're the ones the team's DNA isn't adapted to detect.

---

8. **The hybrid problem**

Some studios genuinely combine families. These are the teams people mythologize: excellent feel plus systems depth, or story

craft plus strong action, or social design plus tactile delight.

But hybrid DNA is rare and fragile. The families' values fight each other. Strong narrative control resists systemic freedom. Competitive clarity resists simulation richness. Content scale resists handcrafted elegance. Live-service needs resist artistic closure.

The studios that pull it off usually have multiple strong subcultures that coexist in productive tension. They haven't eliminated the conflict between families. They've built an organizational structure that lets the conflict resolve into design decisions rather than political warfare. The feel people and the systems people argue, and the argument produces a better game instead of a worse culture.

When the structure fails — when one family's values dominate and the others are tolerated but not empowered — the hybrid collapses into a single-family studio wearing a hybrid costume. The game ships with excellent combat and incoherent systems, or deep systems and hostile onboarding, or vast content and no emotional peak. The missing family's contribution is present as a feature list but absent as a lived instinct. Players sense this even when they can't name it.

---

9. **The broader thesis**

There is no single competency called "making good games." There are multiple institutional adaptations, each fitted to different forms of play, different audiences, and different production truths.

A studio's DNA doesn't determine its ceiling, but it describes the problems the team has learned to solve — and by extension, the problems it hasn't. When a team moves into new territory, the DNA comes along and does what it was trained to do, whether or not that's what the new project needs.

The useful shorthand: *they are organized to solve a particular class of game-design problems.* Everything else follows from noticing which class that is.

---

10. **Related essays**

- *The Studio Primitive: Promise → Proof → Cut → Ship (https://geroldss.github.io/posts/studio-os/)* — the operating loop that makes any DNA family's strengths testable.
- *Making the Thing (https://gerolds.github.io/posts/making-the-thing/)* — how to find a governing commitment and protect it through production.
- *The Dream That Won't Compound (https://gerolds.github.io/posts/the-dream-that-wont-compound/)* — what happens when the operating system prevents the DNA from expressing itself honestly.
- *Finding the Commitment (https://gerolds.github.io/posts/finding-the-commitment/)* — why sticking with a direction matters more than picking the right one.

Drafting assistance: Claude. All claims mine; errors my responsibility.