The Coherence Tax

Series: The Promise Framework — Keeping Game Vision Coherent from Pitch to Live Ops

---

The Coherence Tax: Why Games Break Between “Promise” and Reality

A leadership-friendly framework for keeping vision, production, marketing, and live ops aligned—without requiring a blood sacrifice to the roadmap gods.

If you’ve shipped games long enough, you’ve heard some version of this sentence:

> “This isn’t the game you sold me.”

Sometimes it arrives as a Steam review. Sometimes it’s a Reddit post with 12,000 upvotes. Sometimes it’s your community manager politely translating it into corporate-friendly language like: *“Players feel a mismatch between expectations and experience.”* (Bless them.)

And sometimes—if you’re really lucky—it arrives in a meeting… from your own exec team, after the first retention report.

What’s happening in that moment isn’t just a marketing problem, or a production problem, or a “players don’t understand” problem.

It’s a **coherence problem**.

And coherence, in game development, is expensive.

Welcome to the **Coherence Tax**.

The Coherence Tax (and why leadership should care)

A game’s lifecycle is basically a relay race where everyone runs on different terrain:

1. Early concept and vision: sprinting on hope and caffeine
2. Production: trudging through reality and technical debt
3. Marketing: building desire in a world where attention lasts 0.8 seconds
4. Launch: the universe rolls a D20
5. Live ops: you become a seasonal content factory… with feelings

In that relay, coherence is the baton. Drop it once and you don't just lose time—you lose trust.

The Coherence Tax shows up as:

- **Acquisition waste**: you get installs… from the wrong expectations
- **Early churn**: players leave before the game has a chance to be itself
- **Narrative damage**: once the story becomes "broken promise," it's hard to rewrite
- **Roadmap chaos**: post-launch becomes "catch up to what people thought we shipped"
- **Team morale hits**: nothing drains teams faster than "fix the perception gap" work
- **Stakeholder whiplash**: everyone has a solution, none share a map

If you lead teams, budgets, or outcomes, this matters because coherence isn't a soft concept. It's a multiplier on every KPI you care about.

So the real question is:

How do you keep a game coherent from pitch… to launch… to year three?

That's what this series is about.

The part we don't say out loud: coherence is market fit

Let's steal a concept from product thinking, but keep it practical:

Market fit isn't just "lots of people bought it."
It's "the right audience shows up—and then stays—because the experience matches what they expected."

That's exactly where coherence lives:

- If you attract the wrong expectations, **you burn marketing spend**.
- If you attract the right expectations but don't deliver early, **you burn retention**.
- If you deliver at launch but drift over time, **you burn trust**.

So if you want a durable game, you need a durable way to define—and govern—what "fit" actually means.

That tool is the **Promise**.

Introducing the Promise (the version you can actually use)

Studios already have tools that orbit this problem:

- vision statements
- pillars
- core loop definitions
- “core fantasy”
- USP positioning
- strategy decks that are 80% gradients

They help—until they don’t. Especially once different departments interpret them differently.

Here’s the construct that ties those tools together in a way leadership can govern:

The Promise = Needs + Fantasies *(for a target player)*

Promise is the commitment your game makes to a **specific audience**, expressed as:

- **Needs**: *why they care* (motivation)
- **Fantasies**: *what they want to inhabit* (desire)

If your Promise is clear, it becomes a shared language across design, production, marketing, and live ops.

If your Promise is fuzzy, your organization will fill in the blanks… creatively.
(“Creatively” is leadership code for “expensively.”)

“Players don’t buy features.” (They buy meaning.)

Here’s a quick test.

Ask five stakeholders what your game is.

If you get answers like:

- “It’s a roguelite with crafting.”
- “It’s a social sandbox.”
- “It’s a narrative RPG with systemic AI.”
- “It’s a live-service action game.”
- “It’s like *X* meets *Y* but with more *Z*.”

…you might have a genre. You might have a feature set. You might even have a very impressive slide.

But do you have a **Promise**?

Players don’t show up for “crafting.” They show up for what crafting **means**:

- mastery
- self-expression
- control
- status
- belonging
- the satisfaction of “I built this”

Same mechanic, different meaning. That’s why coherence is hard—and why it needs a framework that sits above features.

Expectation debt (the quiet killer)

Most studios track technical debt. Fewer track **expectation debt**.

Expectation debt is what you accrue when you generate strong anticipation—then don’t have enough playable proof to back it up.

The tricky part?

You can accumulate expectation debt **even when everyone is acting in good faith**:

- Marketing builds the strongest narrative they can from what they’re given.
- Leadership wants a bold position.
- Production is doing its best under constraints.
- Designers want to ship the best experience possible.

Expectation debt doesn’t require malice. It requires mismatch.

And mismatch is what the Promise is designed to prevent.

What a Promise is (and what it isn’t)

Let’s make this extremely practical.

A Promise is:

- a compact statement you can repeat across meetings
- anchored in a target player’s needs and fantasies
- stable enough to guide decisions over time
- specific enough to be testable in the first minutes of play

A Promise is not:

- a slogan
- a genre label
- a mood board
- a feature list
- “we’re making the best game in the category” (everyone is, somehow)

If your Promise can’t survive contact with production tradeoffs—scope cuts, staffing changes, platform constraints—it’s not a Promise. It’s a wish.

A one-minute history (because we keep reinventing this)

Every generation of teams invents a new name for “the thing we must not lose.”

- In one era it’s a **pitch**.
- Then it’s **pillars**.
- Then **core loop**.
- Then **core fantasy**.
- Then “the one thing we must prototype early.”
- Then live ops shows up and says: “Cool. Now do it every eight weeks forever.”

Different labels. Same function: **align what we tell players to expect with what the game reliably enables.**

The Promise is simply that same tool, formalized as:

- player-centered (needs + fantasy),
- legible across departments,
- usable for decisions—not just presentations.

The leadership use case: decisions, not poetry

Here’s where this starts paying rent.

A Promise gives leadership a decision filter. Instead of debating “do we like this idea?” you ask:

Does this directly reinforce the Promise to the target player?

That question does two things instantly:

1. It moves the debate from internal taste to external value.
2. It gives you permission—politely—to say no.

If you've ever led a production with multiple stakeholders, "permission to say no" is basically a superpower.

Quick interactive moment (yes, right now)

Think of your current project—or your last one.

Answer these three questions:

1. **Who is the target player we're building for?**
   (Not "everyone," unless your budget is also "infinite.")
2. **What need are we satisfying for them?**
   (Mastery? Autonomy? Social belonging? Self-expression? Status? Comfort?)
3. **What fantasy are they stepping into?**
   (Who are they? Where are they? What do they do?)

If you can answer those, you can draft a Promise.

If you can't, don't worry—you're not alone. Most teams can talk mechanics for hours and struggle to name the underlying need in a sentence.

That's why we're formalizing it.

What's coming next in this series

This first article is the frame: **coherence is a lifecycle problem**, and **Promise** is the tool that makes it governable.

Next, we'll make the Promise concrete and usable:

- **Post 2:** *Anatomy of a Promise — Needs + Fantasies* *(and why players don't buy features)*
- **Post 3:** *The Missing Link — Player Projects* *(how Promises become play)*
- **Post 4:** *Promise as Governance — the decision filter that helps you say "no"*
- **Post 5:** *Keeping the Promise — live ops, monetization, and co-creation*
- **Post 6:** *Operationalizing it — the Integrity Grid + metrics dashboard + casebook*

If you're reading this as a producer or exec: you'll get templates you can use in milestone reviews, not just theory.

Speaking of templates…

Tool: The One-Sentence Promise Draft (v0)

Here's the simplest usable form:

Promise template

For `[target player]`,
this game delivers `[need(s)]`
through the fantasy of `[setting + character + action]`.

If you want the executive cut, keep it to one breath. If you need oxygen halfway through, it's two Promises hiding in a trench coat.

Examples (generic on purpose)

- **For** players who love mastery and earned power,
  this game delivers competence and status,
  through the fantasy of becoming a relentless hunter in a hostile world—tracking, preparing, and winning impossible fights.
- **For** players who want calm autonomy and self-expression,
  this game delivers control and creativity,
  through the fantasy of building a life in a cozy community—growing, decorating, and shaping relationships at your own pace.
- **For** players who crave social belonging and high-stakes competition,
  this game delivers relatedness and status,
  through the fantasy of leading a squad in tactical battles—coordinating, outsmarting, and climbing together.

Leadership prompts (use these in meetings)

- "If we cut 30% scope, what part of the Promise must still be true?"
- "What does the player get to *do* in the first 10 minutes that proves this Promise?"
- "Which department is currently interpreting the Promise differently—and what's it costing us?"

Homework (10 minutes)

Write three versions:

1. the marketing-friendly Promise
2. the production-safe Promise
3. the player-true Promise

The goal isn't that they match on day one. The goal is that the conversation becomes explicit—before the community does it for you.

If you only do one thing this week

Draft the one-sentence Promise and read it out loud in your next leadership meeting.

If people immediately start adding clauses, congratulations: you just discovered where coherence is at risk.

Next post, we’ll unpack the two moving parts—**Needs** and **Fantasies**—and turn this into something you can use as a prioritization tool, not just a nice sentence.

Next post: **Anatomy of a Promise: Needs + Fantasies**

All trademarks and registered trademarks are the property of their respective owners. This article is not affiliated with or endorsed by any of them.

Anatomy of a Promise

Series: The Promise Framework — Keeping Game Vision Coherent from Pitch to Live Ops

---

Anatomy of a Promise: Needs + Fantasies

Or: why players don’t buy features, and why your studio can’t afford to sell vibes.

In the first post, we talked about the **Coherence Tax**—the expensive surcharge you pay when what players *expect* and what your game *delivers* drift apart.

Today we’re building the antidote.

Not a 60-slide “positioning narrative.”
Not a mood board (though… we all love a mood board).
Not a slogan.

A tool you can use in meetings without anyone needing to pretend they read the whole doc.

Quick recap: what a Promise is

A **Promise** is the commitment your game makes to a **target player**, expressed as:

Promise = Needs + Fantasies

- **Needs** = why they care (motivation)
- **Fantasies** = what they want to inhabit (desire)

The leadership payoff is simple: **Needs tell you what to prioritize. Fantasies tell you what to align.**
And yes—those are different problems.

Part 1 — Needs: the engine of motivation (the “Why”)

The moment you say “player needs,” half the room expects a psychology lecture. The other half expects a monetization debate. (Both groups are, unfortunately, correct.)

But we can keep this extremely practical:

Needs are not academic trivia

In a leadership context, needs are:

- **segmentation** (who this is for)
- **retention design** (why they stay)
- **decision filters** (what matters when scope collapses)

If you don’t name the need, you’ll accidentally design for “every need,” which is how you end up shipping a buffet where nobody is hungry.

The short list leaders actually need

There are many models. You don’t need to pick a religion. You need to pick priorities.

Most needs you’ll design for cluster around:

- **Autonomy**: “Let me choose. Let me own my approach.”
  Signals: build choices, multiple solutions, player-driven goals.
- **Competence**: “Let me improve. Let me earn power.”
  Signals: mastery loops, skill expression, meaningful challenge, clear feedback.
- **Relatedness**: “Let me belong.”
  Signals: co-op friction reduction, guild identity, shared rituals, social status.

And you’ll see these constantly in commercial reality:

- **Status / Esteem**: “Notice me.”
  Signals: rankings, rare rewards, visible accomplishments.
- **Self-expression / Creativity**: “Let me make it mine.”
  Signals: customization, building, fashion, roleplay.
- **Comfort / Escape**: “Give me a place to breathe.”
  Signals: soothing rhythms, low-stress mastery, predictable progress.

The leadership trap: mistaking mechanics for needs

A producer says: “We need crafting.”
A designer says: “We need deeper combat.”
Marketing says: “We need more spectacle.”
An exec says: “We need a battle pass.”

Those are not needs. Those are **tactics**.

A need is:

- “Players want competence—earned mastery.”
- “Players want relatedness—belonging and shared identity.”
- “Players want autonomy—freedom to set goals.”

Mechanics should be chosen *because* they reliably satisfy the need **inside your fantasy**.

If you want to end a circular meeting, try this sentence:

> “Which player need does that serve, and how will we prove it in the first hour?”

It’s amazing how quickly everyone becomes allergic to vague ideas.

Operationalizing needs (without turning this into a PhD)

Here’s the practical truth: teams often use “motivation models” as working tools—especially for market research and competitor mapping.

You’ve probably seen or used frameworks like:

- **Quantic Foundry**-style motivation clusters (mastery, competition, creativity, immersion, etc.)
- **GameRefinery**-style motivational drivers and player archetypes

You don’t need to enshrine any one model. The point is what they enable:

- Map **target audience needs**
- Compare **competing products**
- Identify **underserved intersections** where your fantasy can land

Leadership translation:
A **needs framework** turns “we think players will like this” into “this audience reliably chases these Projects.”

And Projects are where retention lives (next post, but you knew I’d say that).

Part 2 — Fantasies: the engine of desire (the “What”)

If needs are the “why,” fantasies are the “what.”
But fantasy here doesn’t mean dragons. It means:

Fantasy = the role the player imagines themselves inhabiting

It’s what makes the need *desirable.*

A competence need can be satisfied by a spreadsheet.
A competence need inside the fantasy of “ace pilot” is… a game.

A simple way to describe fantasy (that your whole studio can share)

Fantasy breaks cleanly into three buckets:

1. **Setting** — *Where am I?*
2. **Character** — *Who am I?*
3. **Action** — *What do I do?*

That’s it. That’s the whole trick. It’s simple enough that:

- marketing can message it,
- art can visualize it,
- design can systematize it,
- production can plan around it,
- leadership can govern it.

Example (generic)

- Setting: “a collapsing space colony”
- Character: “a scavenger with a questionable past”
- Action: “sneak, bargain, and improvise your way to survival”

Now you can ask alignment questions without a philosophy debate:

- Does our art reinforce the setting fantasy?
- Does our UX reinforce the character identity?
- Does our core loop reinforce the action fantasy?
- Does our marketing emphasize the same three things?

If those answers diverge, coherence starts leaking immediately.

Meta-fantasies: “It feels like X” (use with caution)

Sometimes your Promise isn’t just “be a stealthy assassin.” Sometimes it’s:

- “Console-quality open world… on mobile.”
- “A true Diablo experience, but on your phone.”
- “The nostalgic feeling of that classic you loved—modernized.”

That’s a **meta-fantasy**: a referential Promise that borrows meaning from a shared library of past experiences.

These can be incredibly powerful… and incredibly risky.

Why? Because you inherit expectations you don’t fully control:

- what “fair” means
- what “earned power” means
- what “authentic” means
- what the emotional rhythm “should” feel like

Leadership note: meta-fantasies aren’t bad. They just require discipline—especially around monetization and early validation.

Putting it together: Promise = Needs + Fantasies

Here’s the synthesis:

- **Needs** tell you what psychological outcome must reliably happen.
- **Fantasies** tell you the narrative wrapper that makes that outcome compelling.
- Together, they form a Promise you can:
  - communicate externally (marketing),
  - enforce internally (governance),
  - and sustain long-term (live ops).

A Promise without needs becomes “cool stuff.”
A Promise without fantasy becomes “a feature spec.”

You need both.

Adjacent concepts (so we don’t start a vocabulary war)

You might already use:

- value proposition
- brand promise
- experience goals
- pillars
- core fantasy

Good. Keep them.

This framework doesn’t replace those—it *connects* them by forcing one stable center:

> “What needs are we satisfying for whom, through what fantasy?”

If everyone can answer that the same way, the rest of your tools suddenly work better.

Mini diagram: the Promise Map (highly technical)

```
TARGET PLAYER
   |
   v
NEEDS (Why)  +  FANTASY (What)
            - Setting (Where)
            - Character (Who)
            - Action (What I do)
   |
   v
PROMISE (one sentence everyone can repeat)
```

Yes, this diagram fits in a producer’s notebook. That is the point.

Workshop: Promise Mapping in 45 minutes (leadership-friendly)

This works with a mixed room: production, design, marketing, maybe a brave executive.

Step 1 — Pick the target player (10 minutes)

Not “everyone.” Be specific enough that you can picture a real person.

- What are they trying to feel?
- What do they avoid?
- What do they brag about?

Step 2 — Choose 2–3 primary needs (10 minutes)

If you pick five, you’re building multiple games.

Helpful framing:

- Primary need: what must happen repeatedly for the game to work
- Secondary need: what strengthens it
- Optional need: what’s nice but not required

Step 3 — Define fantasy themes (10 minutes)

Fill in:

- Setting:
- Character:
- Action:

If you can’t define the action, you don’t have a gameplay Promise yet—only a world pitch.

Step 4 — Draft the one-sentence Promise (10 minutes)

Use the template:

For `[target player]`,
this game delivers `[need(s)]`
through the fantasy of `[setting + character + action]`.

Step 5 — The scope stress test (5 minutes)

Ask: **If we cut 30% scope tomorrow, what must survive for the Promise to remain true?**

Those survivors become your **non-negotiables.**
Everything else becomes:

- later,
- smaller,
- or removed.

This step alone prevents a shocking number of roadmap tragedies.

Tool: The Promise Map (one-page template)

Copy this into your doc, Miro, Figma, Notion, or the back of a napkin (the traditional producer medium).

1) Target player

- Who is this for (not everyone)?
- What are they trying to feel?

2) Needs (pick 2–3)

- Primary:
- Secondary:
- Optional:

3) Fantasy themes

- **Setting (Where):**
- **Character (Who):**
- **Action (What do I do):**

4) The Promise (one sentence)

For ______, this game delivers ______ through the fantasy of ______.

5) Non-negotiables (what must be true)

- Must be true that…
- Must be true that…

6) Anti-Promise list (things we should *not* add)

This is the secret weapon. Write the temptations down.

- “It’s cool but it dilutes the fantasy.”
- “It satisfies a different need.”
- “It belongs to another game.”

7) Early proof (FTUE)

- What happens in the first 5 seconds to 10 minutes that proves the Promise is real?

If you only do one thing this week

Run the 45-minute Promise Mapping workshop with one representative each from:

- production
- design
- marketing

Then do the scope stress test.

If you can’t agree on non-negotiables, you’ve just found where coherence will break later—while it’s still cheap to fix.

Next up

Now that we can build a Promise, we need the missing link that makes it playable.

Because here’s the uncomfortable truth:

Promises don’t retain players. Projects do.

Next post: **The Missing Link — Player Projects (How Promises Become Play).**

All trademarks and registered trademarks are the property of their respective owners. This article is not affiliated with or endorsed by any of them.

The Missing Link

Series: The Promise Framework — Keeping Game Vision Coherent from Pitch to Live Ops

---

The Missing Link: Player Projects (How Promises Become Play)

Or: why your trailer can’t retain players, and why your quest list is not the point.

Let’s start with a scene you’ve probably lived through:

You’ve got a solid Promise. Everyone can say it. Marketing is aligned. The deck is gorgeous. The team feels good.

Then you launch, and a chunk of players bounce early with feedback like:

- “I don’t get what I’m supposed to do.”
- “There’s nothing to do.”
- “It feels grindy.”
- “It’s not what I expected.”
- “I wanted to be a *[fantasy]*… but I’m just doing chores.”

This is the moment teams often reach for more content, more quests, more “stuff.”

Sometimes that helps. Often it doesn’t.

Because the real problem usually isn’t “lack of content.”

It’s that the Promise didn’t reliably turn into player Projects.

Projects are the missing link between what you *say* and what players actually *do*.

Recap: Promise is the contract. Projects are the proof.

From earlier posts:

- Promise = Needs + Fantasies (for a target player)
- Fantasies break down into *Setting / Character / Action*
- A strong Promise is testable early and governable over time

Now we add the bridge that makes this whole thing operational:

Project = the plan the player forms to pursue meaning inside your systems.

Not a quest. Not a task list. Not "content."

A Project is player-authored intent.

And that intent is what turns anticipation into engagement.

The Anticipation Bridge: why Projects are where coherence starts

Let's patch in one of the paper's key ideas in plain-English form:

Your Promise enters through anticipation

Marketing, announcements, trailers—whatever your channel—does one important thing:

It makes certain fantasies *salient*.
It tells players, "This is what you'll get to be and do."

So before they ever touch your game, players are already forming a mental shortlist of:

- what they want to try
- what they expect to be possible
- what "success" will feel like

This is why the first hour matters so much: players arrive with pre-built intention.

In-game, players convert anticipation into Projects

When they can see a path to the fantasy, they form Projects like:

- "I'll master stealth and ghost this mission."
- "I'll build the perfect team."
- "I'll climb the ladder."
- "I'll make this place mine."

That's a cognitive shift:

- from *wanting* to *planning*

- from *hype* to *agency*

Validation locks trust (and failure creates dissonance)

When players succeed at meaningful Projects, two things happen:

1. Their Project feels worthwhile (need satisfied).
2. The Promise feels credible ("Yes, I can be that fantasy here.")

When they *can't* pursue the Projects they believed were possible—because systems don't support it, or it's gated, or it's unreliable—players experience a kind of internal mismatch:

> "I expected X, I tried X, the game wouldn't let me."

That mismatch is where "broken promise" narratives are born.

No need to overthink the psychology here. Leadership translation is enough:

Projects are how expectations become behavior.
If Projects can't form, retention can't stabilize.

Why leadership should care about Projects

Because Projects are the unit of engagement.

When players have good Projects, you'll see:

- longer sessions (they're pursuing something)
- stronger retention (they have plans)
- better sentiment (intent matches experience)
- healthier monetization (spending supports meaningful goals)

When players don't form Projects, you'll see:

- "nothing to do" complaints even in content-rich games
- early churn even when the Promise is attractive
- feature thrash: adding systems to "fix retention" without fixing intent formation
- roadmap drift: chasing symptoms instead of the cause

A leadership shortcut:

> If players can't form Projects, they can't inhabit your fantasy.
> If they can't inhabit your fantasy, your Promise becomes marketing fiction.

The Promise → Project loop (the engine of coherence)

Here's the loop we want to protect:

1. Promise activates anticipation
2. Player forms Projects aligned with that anticipation
3. Agency + challenge lets them pursue Projects
4. Rewards validate the Projects (and the Promise)
5. Players form new Projects, deeper and longer-term

Fancy diagram:

```
PROMISE (Needs + Fantasies)

       |

       v

  ANTICIPATION

       |

       v

PLAYER PROJECTS (intent + plan)

       |

       v

AGENCY + CHALLENGE (choices, mastery, social play)

       |

       v

REWARD + VALIDATION (proof the fantasy is real)

       |

       v

 STRONGER BELIEF IN THE PROMISE

       |

       v

  NEW PROJECTS (repeat)
```

Where teams often struggle is thinking they can jump from Promise straight to retention with "content."

But Projects are the actual mechanism.

Project typology (practical, not academic)

Projects vary along three axes that matter for design and leadership decisions.

1) Duration: micro → long → infinite

- Micro (seconds/minutes): “Land this jump.” “Win this duel.”
- Short (5–30 min): “Clear this dungeon.” “Finish this contract.”
- Mid (hours): “Build a viable build.” “Unlock the next tier.”
- Long (tens of hours): “Max my character.” “Reach ranked milestone.”
- Open-ended: “Become a top clan leader.” “Climb forever.”

Leadership takeaway: if your first meaningful Project is a “long Project,” you’re asking for commitment before you’ve paid the player emotionally.

2) Formalization: explicit vs tacit

- Explicit Projects (game-authored): quests, achievements, seasonal milestones
- Tacit Projects (player-authored): fashion builds, speedrunning, “no-hit” runs, roleplay, mastery challenges you never named

Leadership takeaway: tacit Projects aren’t accidents. You can design for them by leaving room for expression, optimization, and mastery depth.

3) Structure: closed vs open

- Closed Projects (“Game”): puzzles, boss fights, defined win conditions
- Open Projects (“Play”): building, experimenting, social rituals, emergent economies

Leadership takeaway: closed Projects give clarity and pacing; open Projects give ownership and longevity. Durable games usually mix both.

Projects are “good” when they satisfy needs and express fantasy

A Project works when it:

1. expresses the fantasy (setting/character/action), and
2. satisfies the need (autonomy/competence/relatedness/etc.)

Example patterns:

- Competence + hunter fantasy:
  “I’ll learn the tells and master timing.” → validates mastery
  vs “I’ll wait for RNG to carry me.” → undermines mastery fantasy
- Relatedness + squad fantasy:
  “We’ll coordinate roles and win together.” → validates belonging
  vs “I’ll avoid other players because grouping is pain.” → signals mismatch

This is how Projects become diagnostic tools.

Where Projects fail (and how you spot it early)

Failure mode 1: Players can’t imagine a plan

Symptoms:

- “I don’t know what to do.”
- wandering without intent
- FTUE drop-offs

Fix:

- present a clear, Promise-dense Project immediately

Failure mode 2: Players imagine a plan… but can’t execute it

Symptoms:

- “Stealth doesn’t work.”
- “Build variety is fake.”
- “Exploration isn’t rewarded.”

Fix:

- strengthen agency; remove blockers to the core fantasy Projects

Failure mode 3: Projects complete, but don’t validate the Promise

Symptoms:

- “It’s fine, but it doesn’t feel like *that*.”
- “Rewards don’t matter.”
- “It’s chores.”

Fix:

- make rewards meaningful in fantasy terms (identity, progression, status, expression)

Failure mode 4: Monetization disrupts Projects

Symptoms:

- “Pay-to-win” narrative
- competence/fairness frustration
- churn among mastery-driven cohorts

Fix:

- monetize in ways that enrich identity or convenience without invalidating mastery Projects
  (We go deeper in Post 5.)

Tool: The Project Inventory (the thing you can use on Monday)

This turns Promise from a sentence into an engagement plan.

Step 1 — List top Projects by horizon

Write Projects as player intentions (“I will…”), not features.

First 5 seconds to 10 minutes (FTUE)

- I will…
- I will…

First hour

- I will…
- I will…

First 3 hours

- I will…
- I will…

First 30 hours / season lifecycle

- I will…
- I will…

Step 2 — Tag each Project

For each Project, tag:

- Need satisfied (autonomy / competence / relatedness / etc.)
- Fantasy theme expressed (setting / character / action)
- Type (explicit vs tacit)
- Structure (closed vs open)
- Duration (micro → open-ended)

Step 3 — Find implied Projects (the “marketing shadow”)

Ask:

- What Projects does our marketing imply?
- Can the game support them reliably?
- Are they available early enough to prevent expectation debt?

If marketing implies “be a stealth god” and stealth isn’t viable until hour 8, you’ve created a coherence trap.

If you only do one thing this week

Run a Project Inventory with design + production + marketing in one room.

If marketing implies Projects the game can’t support, don’t argue about messaging—fix either the product or the Promise.

Because Projects are where coherence lives or dies.

Next up

Now we have Promises and Projects.

Next we weaponize them for leadership:

Next post: **Promise as Governance: the decision filter that helps you say “no.”**

All trademarks and registered trademarks are the property of their respective owners. This article is not affiliated with or endorsed by any of them.

Promise as Governance

Series: The Promise Framework — Keeping Game Vision Coherent from Pitch to Live Ops

---

Promise as Governance: The Decision Filter That Helps You Say “No”

Or: how to keep your project coherent when everyone has a “small, reasonable request.”

If you’ve ever led a game project, you’ve lived this movie:

A stakeholder walks in with an idea that is, objectively, not terrible.

Marketing wants a new beat for the trailer.
Design wants to “just add” a system.
Live ops wants a mode that will “definitely boost engagement.”
Platform wants a feature for featuring.
Leadership wants a monetization hook “because the forecast.”

Individually, each request makes sense. Collectively, they form a majestic creature known as:

The Roadmap Hydration Monster
(It’s always thirsty. It never gets smaller.)

This is where projects lose coherence—not because anyone is incompetent, but because the decision process has no shared, enforceable center.

That center is the **Promise**.

And yes: we’re going to use it as governance.

Recap: the framework, in one breath

- **Promise = Needs + Fantasies** (for a target player)

- **Projects** are how players actualize that Promise through agency
- Governance means keeping that chain coherent across:
    - scope changes,
    - stakeholder pressure,
    - production reality,
    - marketing narratives,
    - live ops evolution

Today is the most practical part:

How the Promise becomes your decision filter.

The uncomfortable truth: “Vision” is not enforceable

Vision statements tend to be:

- inspirational,
- abstract,
- and interpreted differently by every department.

That’s not a criticism. That’s what vision is for.

But governance needs something else—something that can answer:

> “Should we do this feature?”
> “Should we message this?”
> “Should we cut this?”
> “Should we run this event?”
> “Should we monetize this way?”

That’s why the Promise matters. It’s player-centered, testable, and legible across teams.

The governance question

Instead of debating taste or politics, ask:

Does this directly reinforce the Promise to the target player?

If the answer is “no,” it’s not automatically a bad idea.

It’s just not *this game’s* idea.
(And yes, it can still live in the sequel pitch deck with all the other “great ideas we didn’t have time for.”)

The governance artifact you actually need: non-negotiables

Here's the leadership move that saves projects:

Derive "non-negotiables" from the Promise

Non-negotiables are the smallest set of truths that must remain true for the Promise to still be honest.

They're not features. They're **outcomes**.

Example pattern:

- Need: **competence**
- Fantasy: **master assassin**
- Non-negotiable: "Stealth must be reliable and expressive early. The player must feel clever, not clumsy."

Another:

- Need: **relatedness**
- Fantasy: **squad leader**
- Non-negotiable: "Co-op must be frictionless and rewarding. Group play can't feel like paperwork."

Another:

- Need: **autonomy**
- Fantasy: **build-your-own life**
- Non-negotiable: "Player choice must matter. The game can't constantly yank control back."

Why this matters under pressure

When schedule pressure hits (it will), non-negotiables tell you what to protect.

Without them, you cut randomly:

- you keep "content" and cut the feel,
- you keep features and cut the fantasy,
- you keep systems and cut the part players came for.

Non-negotiables let you cut scope without cutting identity.

FTUE is not "tutorial." It's "Promise proof."

If your Promise is “be the thing,” the first minutes of play must let the player *begin being the thing*.

Not “read about the thing.”
Not “wait until hour 6 for the thing.”
Not “here’s your first stick—someday you’ll have the lightsaber.”

A leadership rule of thumb

If a player can’t form a Promise-dense Project in the first 5–10 minutes, you’re accumulating expectation debt.

And expectation debt is expensive post-launch.

So FTUE isn’t “polish.” It’s a credibility milestone.

Prototyping the Promise: don’t sell what you can’t play yet

Different studios use different labels for this, but the principle is timeless:

You need early playable proof of the Promise

Not a pitch. Not concept art. A slice where the fantasy is real.

If your Promise is “master assassin,” you need:

- stealth that works,
- traversal that feels empowering,
- a moment that validates identity.

If your Promise is “cozy autonomy,” you need:

- early agency,
- a soothing rhythm,
- a quick feeling of ownership.

If your Promise is “competitive squad mastery,” you need:

- meaningful roles,
- coordination moments,
- outcomes that reward teamwork.

Leadership implication:
If you can’t demonstrate Promise-dense play early, you don’t have a stable basis for marketing scale or long-term roadmap confidence.

That’s not negativity. That’s risk management.

Handling stakeholder requests without becoming the villain

Stakeholders aren’t the enemy. Misalignment is.

Here’s a polite, repeatable response that saves your sanity:

> “Love the idea. Let’s test it against the Promise.”

Then use this mini scorecard:

The Promise Impact Questions

1. **Which need does this strengthen for our target player?**
2. **Which fantasy theme does this reinforce (setting/character/action)?**
3. **Which player Projects does it enable or deepen?**
4. **What does it risk diluting or contradicting?**
5. **Can we prove it in the first hour of play?**

If a request can’t answer those, it’s probably:

- a different audience,
- a different Promise,
- or a future expansion.

That’s not rejection. That’s coherence management.

Soft humor you can use in meetings

- “This is a great idea… for our next game.”
- “That feature is amazing. It also eats our Promise for breakfast.”
- “We can add it. Which Promise are we removing to make room?”

Deliver with kindness. And maybe snacks.

When you should change the Promise (yes, sometimes you should)

Governance isn’t stubbornness. Sometimes the market, the product reality, or the team’s strengths reveal a better Promise.

Consider revisiting the Promise when:

- your most retained cohort is pursuing a different Project ecosystem than you planned

- your strongest “fun” consistently lives outside the original Promise
- production constraints make the original Promise unreliable
- live ops reality shifts the identity of the game over time

The key is: **make the change explicit.**

Silent Promise drift is the worst state:

- marketing sells one Promise
- the game rewards another
- live ops pushes a third

That’s how you end up with three internal truths and one confused audience.

Tool: The Promise Review (30-minute meeting format)

Run this at:

- greenlight
- vertical slice
- alpha/beta
- pre-launch marketing lock
- every seasonal planning cycle

Attendees (keep it small)

- production (owner)
- design (core loop owner)
- marketing/brand (promise articulation)
- live ops (if applicable)
- a leadership stakeholder (decision authority)

Agenda (30 minutes, timer enforced)

1) Restate the Promise (2 min)

Read it out loud. One sentence. No edits in real time.

2) Restate the top Projects (5 min)

- first 10 minutes: top 1–2 Projects
- first hour: top 2–3 Projects
- long-term: top 2 Projects

3) Review what changed since last check (10 min)

For each major change request:

- does it reinforce the Promise?
- which Projects does it strengthen?
- what does it dilute?

4) Decide: protect, cut, or defer (10 min)

Use non-negotiables as the shield:

- Protect: strengthens Promise/Projects
- Cut: dilutes or contradicts
- Defer: valuable but not now

5) Confirm one action (3 min)

- “We will adjust X to strengthen Y Project by next milestone.”

That’s it. No extra slides required.

If you only do one thing this week

Run one Promise Review with your leadership team.

Then write down:

- your **non-negotiables**
- your **anti-Promise list** (the temptations you must resist)

Because saying “no” is easiest when you’ve already agreed what “yes” means.

Next up

Governance keeps you coherent at decision time.

But games don’t end at launch. They live. They evolve. They get monetized. Communities reinterpret them. Updates can strengthen or quietly betray what players came for.

Next post: **Keeping the Promise Over Time: Live Ops, Monetization, and Co-Creation**

All trademarks and registered trademarks are the property of their respective owners. This article is not affiliated with or endorsed by any of them.

Keeping the Promise over time

Series: The Promise Framework — Keeping Game Vision Coherent from Pitch to Live Ops

---

Keeping the Promise Over Time: Live Ops, Monetization, and Co-Creation

Or: how Promises die quietly in Season 3—and what leadership can do about it.

If launch is where your Promise gets judged…

Live ops is where your Promise gets **audited**.

Relentlessly.
By millions of players.
Who are extremely motivated to notice when you change the deal.

And here’s the uncomfortable part: most Promise breaks don’t happen in a big dramatic explosion. They happen in small, “reasonable” decisions over time:

- an event that doesn’t match the fantasy
- an economy tweak that changes what “skill” means
- a content drop that pulls players into a different identity
- a monetization feature that turns progress into a purchase decision
- a seasonal meta that rewards behaviors the Promise didn’t sign up for

It’s rarely malicious. It’s usually… *Tuesday.*

So today we’ll cover three leadership problems:

1. How to renew Projects without diluting the Promise (live ops coherence)
2. How monetization can strengthen or sabotage the Promise (trust management)
3. How the community co-authors the Promise whether you like it or not (co-creation)

And we’ll end with a checklist you can run before approving a season.

Quick recap: why live ops is a Promise problem

From earlier posts:

- **Promise = Needs + Fantasies**
- **Projects** are how players turn that Promise into action (and retention)
- Retention comes from a renewable ecosystem of Projects that validate the Promise

Live ops, at its best, is:

> **A factory for fresh Projects that still feel like "this game."**

At its worst, it becomes:

> "We're throwing content at the DAU graph and hoping the fantasy doesn't notice."

Players notice.

---

1) Live ops coherence: renewing Projects without breaking identity

The real output of a season is not "content"

It's **new Projects**.

Think in player intention statements:

- "This season, I will climb the new ladder."
- "I will master the new weapon."
- "I will complete the collection."
- "I will lead my group through the new raid."
- "I will express myself with the new cosmetics."
- "I will return every week because there's a ritual now."

If your season doesn't create Projects players want to adopt, it's just… assets.

The "Promise drift" pattern

Most Promise drift happens when live ops starts optimizing for a different need than the game originally promised.

Example pattern:

- The Promise is about **competence** (mastery, fair challenge).
- But a season leans heavily into **chance** (RNG-heavy events, loot roulette, power spikes).

- Players feel a values conflict: “This isn’t mastery anymore.”

Or:

- The Promise is **social belonging** (squad identity, teamwork).
- But updates reward solo play more efficiently.
- Now the game’s best Projects are misaligned with its social fantasy.

The fix is not “don’t innovate.” The fix is: **innovate inside the Promise.**

A leadership rule of thumb

If a seasonal feature becomes the dominant way to play, it must be Promise-consistent.
Because dominance rewrites identity.

Side mode? Fine.
Main loop? That’s the game now.

---

2) Monetization alignment: the trust multiplier (or the trust shredder)

Let’s say this plainly:

Monetization doesn’t just change revenue.
It changes what the game *means*.

It changes:

- what counts as “earned”
- what counts as “fair”
- what counts as “skill”
- whether time investment feels respected
- whether the fantasy feels authentic

That’s why monetization is a Promise topic, not a finance-only topic.

Monetization that strengthens the Promise

Monetization tends to feel aligned when it:

- **enriches identity fantasy** (cosmetics, personalization, expression)
- **supports Projects without invalidating them** (convenience that doesn’t erase mastery)
- **feels optional** relative to core need satisfaction
- **preserves the values of the fantasy** (earned power stays earned)

In other words:

- Players can still complete meaningful Projects through play
- Spending enhances the journey, not buys the destination

Monetization that undermines the Promise

It becomes dangerous when it:

- bypasses competence ("why learn when you can pay?")
- shifts competitive fairness ("skill matters until wallets arrive")
- blocks core Projects behind paywalls
- turns progression into frustration designed to be purchased away

The result isn't just bad sentiment. It's a Promise rupture:

> "I thought this was about mastery / fairness / earned power… but it's about spending."

Once players reinterpret your Promise as "they're selling power," it's very hard to rebuild trust—because it's a values problem, not a tuning problem.

---

3) Co-creation: the community will reinterpret your Promise—so listen

Players don't just consume your game. They **co-author its meaning**.

They invent:

- new playstyles
- meta strategies
- social rituals
- roleplay norms
- unofficial Projects you never wrote down

Sometimes this is wonderful. Sometimes it's… terrifyingly efficient.

Community Projects are signal

When a community invents a new dominant Project, it's telling you:

- "This is what your systems actually support best."
- "This is what we find meaningful."
- "This is the fantasy we're inhabiting—whether you intended it or not."

Leadership can treat this two ways:

Option A: Fight it

- suppress, nerf, message against it

- sometimes necessary (exploits, harm, fairness)

Option B: Interpret it

- decide whether to embrace it as part of the evolving Promise
- or adjust systems so the original Promise remains dominant

Make it explicit:

Is this emergent Project aligned with our Promise—or is it rewriting it?

Because if you ignore it, the community will choose for you.

When you should pivot the Promise (the grown-up version)

Sometimes live ops reveals a different Promise has stronger pull than the one you planned.

A pivot is worth considering when:

- your most retained cohort is chasing a different fantasy than you marketed
- the strongest “fun” consistently lives outside the intended Promise
- the market response proves a different Promise has stronger pull
- production constraints make the original Promise unreliable at scale

The pivot rule

If you pivot, do it intentionally:

1. **Rewrite the Promise** (needs + fantasies)
2. **Rebuild the Project ecosystem** to match it
3. **Update messaging** so expectations match reality

The worst option is silent drift:

- marketing sells one Promise
- the game rewards another
- live ops pushes a third

That’s not strategy. That’s three products sharing one executable.

Tool: The Live Ops Coherence Checklist (run this before approving a season)

1) Restate the Promise (one sentence)

For ______, this game delivers ______ through the fantasy of ______.

2) List the top Projects the season creates

Write 5–10 “I will…” statements.

3) Map each Project to needs + fantasy themes

For each Project:

- which need does it satisfy?
- which fantasy theme does it reinforce?
- does it deepen the core fantasy—or distract from it?

4) Check for Promise contradictions (danger zone)

Ask:

- is a core Project based on **chance** in a mastery-promising game?
- does it reward **solo efficiency** in a social-promising game?
- does it create **pay pressure** in an earned-power fantasy?
- does it contradict identity (temporary modes that rewrite meaning)?

If yes, either adjust or accept that you’re changing the Promise.

5) Monetization alignment test

- does spending enhance identity or buy outcomes?
- does it accelerate Projects without invalidating them?
- does design introduce frustration to sell relief?
- what story will the community tell about this?

If you don’t know the story, the community will write it for you.

6) Co-creation check

- what tacit Projects are rising?
- what behaviors are becoming dominant?
- are we reinforcing the intended Promise—or rewarding a different one?

7) Decide: reinforce, adjust, or pivot

- Reinforce: strengthens Promise
- Adjust: needs tweaks to avoid contradictions

- Pivot: intentionally redefine Promise

If you only do one thing this week

Before your next season planning lock, run the checklist and ask one brutal question:

“Which Promise are we reinforcing this season—really?”

If the answer is “a different one than our marketing,” you have a choice:

- fix the product to match the Promise, or
- fix the Promise to match the product.

But don’t do neither.

Next up

Now we have Promises, Projects, governance, and long-term coherence rules.

The last step is what leadership always asks for:

“How do we measure this without turning it into KPI soup?”

Next post: **Operationalizing the Promise: The Integrity Grid + Metrics Dashboard + Casebook**

All trademarks and registered trademarks are the property of their respective owners. This article is not affiliated with or endorsed by any of them.

Operationalizing the Promise

Series: The Promise Framework — Keeping Game Vision Coherent from Pitch to Live Ops

1. **The Coherence Tax:** Why Games Break Between "Promise" and Reality
2. **Anatomy of a Promise:** Needs + Fantasies
3. **The Missing Link:** Player Projects (How Promises Become Play)
4. **Promise as Governance:** The Decision Filter That Helps You Say "No"
5. **Keeping the Promise Over Time:** Live Ops, Monetization, and Co-Creation
6. -> **Operationalizing the Promise:** The Integrity Grid + Metrics Dashboard + Casebook

---

Operationalizing the Promise: The Integrity Grid + Metrics Dashboard + Casebook

Or: how to notice Promise drift before your reviews write the postmortem for you.

By now, we've done the hard part: we took "coherence" out of the realm of vibes and turned it into something you can actually govern.

Quick recap of the series so far:

- **Promise = Needs + Fantasies** (for a target player)
- **Projects** are how players turn that Promise into action (and retention)
- **Governance** means using the Promise to say no, prioritize, and protect non-negotiables
- **Live ops and monetization** either renew Projects coherently… or quietly rewrite what the game means

Now the question leadership always asks (and it's a fair one):

> "Great. How do we measure this?"

Not in a "let's add twenty more dashboards" way.
In a "how do we detect Promise breaks early, diagnose causes, and make decisions" way.

We'll do that with three tools:

1. **The Promise Integrity Grid** (a rubric you can use in reviews)
2. **A metrics dashboard that isn't KPI soup** (signals by lifecycle stage)
3. **A casebook with real games** (applied examples, link-free, evergreen)

And we'll end with a one-page playbook you can reuse.

Why “Promise health” needs its own measurement tool

Most studios already measure performance:

- retention curves
- conversion funnels
- ARPDAU / revenue
- session length
- churn

Those are crucial—but they’re symptoms.

Promise failures often look like:

- “marketing worked, but D1 is awful”
- “players bounce after tutorial”
- “reviews are angry even though the game is functional”
- “events spike DAU but tank sentiment”
- “monetization lifts revenue but damages trust”

In other words: you can be “green” on some metrics and still be losing the war—because you’re losing **credibility**.

Promise measurement is about credibility:

Are we delivering the experience we told the target player to expect—and can they form Projects that validate it?

Tool 1: The Promise Integrity Grid (your governance rubric)

Score each dimension 0–2. Total max = 10.

Dimension A — Clarity (0–2)

Can the Promise be stated in one sentence that includes Needs + Fantasy themes?

- **0:** everyone says something different
- **1:** mostly aligned, but fuzzy or multi-Promise
- **2:** crisp, repeatable, shared across departments

Leadership test: ask three people from different departments to state the Promise. If you get three different games, you have a governance problem, not a feature problem.

Dimension B — Early Validation (FTUE) (0–2)

Do the first minutes of play prove the Promise is real?

- **0:** fantasy delayed; first play feels unrelated
- **1:** partial proof; hints but limited agency
- **2:** Promise-dense Projects form immediately

Leadership test: can a player begin “being the thing” in the first 5–10 minutes?

Dimension C — Project Ecosystem (0–2)

Does the game generate a rich ecosystem of Projects that embody the Promise?

- **0:** shallow goals, repetitive chores, weak agency
- **1:** some good Projects, but narrow or fragile
- **2:** multiple overlapping Projects (explicit + tacit, open + closed)

Leadership test: list top Projects at 10 min / 1 hour / 10 hours / endgame. If the list is short, retention will be fragile.

Dimension D — Consistency Over Time (0–2)

Do updates, balance changes, and content drops reinforce—not contradict—the Promise?

- **0:** seasons/events regularly clash with identity
- **1:** mixed; some reinforce, some dilute
- **2:** updates reliably deepen the same fantasy + needs

Leadership test: when players describe “what this game is,” does it match what you intend?

Dimension E — Monetization Alignment (0–2)

Does monetization support meaningful Projects without undermining core needs (competence/fairness/autonomy/relatedness)?

- **0:** spending invalidates Projects or values
- **1:** monetization is tolerated but creates tension
- **2:** monetization strengthens identity and progression without trust damage

Leadership test: can a player pursue core Projects through play without feeling coerced into spending?

How to use the Grid in leadership cadence

Use it at:

- greenlight (Clarity)
- vertical slice (Clarity + Early Validation)
- alpha/beta (Project Ecosystem)
- pre-launch marketing lock (Clarity + Early Validation + Consistency risk)
- seasonal planning (Consistency + Monetization)
- post-update review (Consistency check)

It lets you say:

- “We’re scoring 2 on clarity, 0 on early validation. That’s a launch risk.”
- “We’re shipping features, but our Project ecosystem is thin.”
- “This monetization idea scores 0 on alignment; expect backlash.”

It’s not perfect. It’s useful. That’s the point.

Tool 2: The Promise Metrics Dashboard (without KPI soup)

Track **signals of coherence**, not just raw engagement.

A) Pre-launch: expectation signals (Promise attraction)

You’re measuring whether the Promise is compelling *and understood.*

- Store page conversion / wishlist behavior
- Trailer engagement patterns (completion rate, rewatch, comment themes)
- Community sentiment about *what the game is* (are they describing the same Promise?)
- Creator framing (are they repeating your fantasy themes or inventing their own?)

Signal to watch: hype high + expectations messy = expectation debt.

B) Launch window: credibility signals (Promise proof)

You’re measuring whether the game validates expectations early.

- FTUE completion rate
- Time-to-first meaningful Project (do players form intent quickly?)
- D1/D3/D7 retention (especially new users)
- Early churn reasons (refund tags, surveys, support tickets)
- Review themes: “as advertised” vs “not what I expected”
- Performance/stability issues that block Projects

Signal to watch: strong acquisition + weak D1 is often a Promise mismatch, not a content shortage.

C) Live ops: identity and trust signals (Promise maintenance)

You’re measuring whether updates renew Projects coherently.

- Event participation and return rate
- Retention by cohort segment (do the right players stay?)
- Sentiment shifts around seasons (“doesn’t feel like the game anymore”)
- Monetization narrative indicators (“pay-to-win,” “grindy,” “unfair”)
- Support/social tags around fairness, burnout, confusion

Signal to watch: revenue up + trust down = you’re cashing out the Promise.

Tool 3: Casebook (real games)

Below are examples you can talk about in a room without needing to open a browser.

I’m using the Integrity Grid to show how different failures look—and what leadership should learn.

Kept Promise (AAA): Assassin’s Creed (2007 era positioning)

Promise (simplified): become a master assassin in living historical cities—stealth, traversal, targeted eliminations.
Why it worked: it gives you Projects that directly embody the fantasy (infiltrate, climb, observe, strike, escape), and it lets you start doing “assassin things” early.

Grid (illustrative):

- Clarity: **2**
- Early Validation: **2**
- Project Ecosystem: **2**
- Consistency Over Time: **1–2** (varies by entry, but the core identity remained recognizable)
- Monetization Alignment: **2** (traditional premium model for early entries)
  Total: 9–10

Leadership lesson: your first hour should be a concentrated proof of the identity you marketed.

Kept Promise (Indie): Stardew Valley

Promise (simplified): escape modern grind; build a new life—farming, seasons, relationships, community.

Why it worked: it creates overlapping Projects across horizons (upgrade farm, restore community center, build relationships, explore mines), and it supports both explicit and tacit Projects (self-defined “my perfect farm”).

Grid:

- Clarity: **2**
- Early Validation: **2**
- Project Ecosystem: **2**
- Consistency Over Time: **2**
- Monetization Alignment: **2**
  Total: 10

Leadership lesson: “cozy” isn’t a vibe; it’s a carefully supported Project ecosystem that validates autonomy, comfort, and self-expression.

Kept Promise (Mobile/Service): Genshin Impact

Promise (simplified): explore a vast fantasy world and build a team of characters in a high-production adventure (including the meta-fantasy of “this feels AAA on mobile”).
Why it worked: the long-term Project (“build my team”) is deeply supported, and exploration Projects are continuously renewed.

Grid:

- Clarity: **2**
- Early Validation: **1–2** (varies by player tolerance; core fantasy appears early, depth grows later)
- Project Ecosystem: **2**
- Consistency Over Time: **2**
- Monetization Alignment: **1** (gacha can create tension, but it still supports the “team-building” Project)
  Total: 8–9

Leadership lesson: meta-fantasies (“AAA on mobile”) are powerful—make sure your first hour proves enough of it to avoid expectation debt.

Broken/contested Promise (launch mismatch): No Man’s Sky (at launch)

Promise (simplified): infinite exploration + a sense of shared universe with other explorers.
Failure pattern: the implied social Project (“meet other explorers”) wasn’t supported at launch, creating a sharp mismatch between expectation and reality.

Grid (launch era):

- Clarity: **2**

- Early Validation: **1** (exploration exists, but the shared-universe implication breaks quickly)
- Project Ecosystem: **1**
- Consistency Over Time: **2** (later updates dramatically improved delivery)
- Monetization Alignment: **2**
  Total (launch): 6
  Total (later): higher due to improved consistency and expanded Projects

Leadership lesson: players form Projects from what you imply, not just what you explicitly promise.

Contested Promise (values conflict): Diablo Immortal

Promise (simplified): "authentic Diablo on mobile" (implying earned power through play and loot chase).
Failure pattern: many players perceived monetization as conflicting with the franchise's competence/earned-power values—turning the central Project ("become powerful through grind and mastery") into a spending debate.

Grid (perception-driven):

- Clarity: **2**
- Early Validation: **2** (the surface fantasy is present)
- Project Ecosystem: **2**
- Consistency Over Time: **—** (less the point here)
- Monetization Alignment: **0–1**
  Total: 6–7

Leadership lesson: you can deliver the fantasy "on the surface" and still break the Promise in its values.

Broken competence "feel" Promise: Mighty No. 9 (crowdfunded expectation)

Promise (simplified): spiritual successor to classic precision platform mastery.
Failure pattern: even with visual similarities, execution and feel were perceived as failing the competence/mastery expectation.

Leadership lesson: if the Promise is competence-driven, "feel" is not polish—it's core delivery.

Scope & limits

This series—and the Promise/Project framework—is designed for **commercial game development**, where success depends on lifecycle coherence: acquisition → onboarding → retention → live ops.

It applies best to:

- multi-team productions
- products with marketing and long-term audience expectations
- games where updates, economies, and monetization shape identity over time

It can still be useful for experimental or purely artistic experiences—but the “coherence problem” may not be the goal in those contexts. Some works intentionally subvert expectations. That’s valid. It’s just a different contract.

The one-page Promise Playbook (copy/paste this)

1) One-sentence Promise

For ______, this game delivers ______ through the fantasy of ______.

2) Non-negotiables (3–5 outcomes)

- Must be true that…
- Must be true that…

3) Project Inventory by horizon

10 minutes:

- I will…
- I will…

1 hour:

- I will…
- I will…

10 hours / endgame / season:

- I will…
- I will…

4) Integrity Grid score (0–10)

- Clarity: _/2
- Early Validation: _/2

- Project Ecosystem: _/2
- Consistency Over Time: _/2
- Monetization Alignment: _/2
  Total: _/10

5) Current risks (what might break the Promise next)

- Risk:
- Risk:

6) Next action (one decision, one owner)

- Action:
- Owner:
- When:

If you only do one thing this week

Before your next season planning lock, run the checklist and ask one brutal question:

“Which Promise are we reinforcing this season—really?”

If the answer is “a different one than our marketing,” you have a choice:

- fix the product to match the Promise, or

- fix the Promise to match the product.

But don’t do neither.

Last Words

Thanks for reading—and for caring about coherence (your team will thank you later, quietly, by not having to “fix the narrative” at 2 a.m.). If you try any of the tools from this series, I’d love to hear what worked, what didn’t, and what you’d add to the framework.
