American Impact Review

Theoretical Article

Smart Lean in Gamedev: Reconfiguring the Production Flow When AI Handles Routine and Humans Stop Waiting

Maksim Kananovich*

ITGROUP L.L.C-FZ

* Corresponding author

OPEN ACCESS

Citation:
Maksim Kananovich (2026). Smart Lean in Gamedev: Reconfiguring the Production Flow When AI Handles Routine and Humans Stop Waiting. *Am. Impact Rev.* 10.66308/air.e2026048

Received: June 1, 2026
Accepted: June 7, 2026
Published: June 9, 2026

DOI:
10.66308/air.e2026048

ISSN: 3071-124X

Abstract

This paper introduces the Smart Lean concept, an adaptation of Lean production principles to modern game development built around hybrid teams of human specialists and AI agents. Its core argument is that the dominant waste in game production is not task complexity but waiting time at the handoffs between roles waste that AI tools relocate rather than remove unless the value stream itself is redesigned. Smart Lean reorganizes production into hybrid cells, where AI handles draft and routine work under a Plan Do Check Act cycle while human specialists retain final acceptance and creative ownership. It transforms Value Stream Mapping from a static diagram into a live digital twin that surfaces bottlenecks in real time, converts handoffs between cells into stateful relays carrying quality certificates and context logs, replaces monthly retrospectives with data-driven Kaizen pulled directly from build telemetry, and applies Smart 5S to manage the clutter generated by AI at scale. Drawing on twelve years of implementations across fifteen studios ranging from 20 to more than 500 people, the methodology reportedly cut waiting time from 50-70% to 10-20% of cycle time and shortened production cycles severalfold, shifting specialist value away from routine execution toward decision-making, creative supervision, and flow management.

Keywords: Smart Lean, Value Stream Mapping, digital twin, PDCA, live-ops

1. THE PROBLEM: GAMES HAVE GROWN, BUT CONVEYORS HAVE NOT

Over the past ten years, games have become bigger. Much bigger. Open worlds, hundreds of NPCs, dialogue trees, progression systems, and live operations. But the processes in the studios haven't changed. It's still the same cycle: idea → design document → art → integration → QA → rework. At every handoff, the artifact waits. Sometimes days, sometimes weeks.

We measure development speed in sprints. But the real enemy is not task complexity - it is muda(waste) at the boundaries between roles.

With the arrival of AI tools, many studios rushed to automate everything. Generate textures, write code stubs, assemble builds. This produced local speed gains. But the problem did not disappear - it moved. Now an artist waits not for a colleague but for AI output review. An engineer does not wait for code but drowns in its review [2].

Smart Lean is not about adding AI. It is about redesigning the value stream itself so that AI and humans work as a single cell, not as links in a long chain of waiting.

2. WHERE THE METHODOLOGY COMES FROM: 12 YEARS IN LIVE STUDIOS

Nothing in this paper was born in academic isolation. It was extracted from real projects: mobile games, mid-core, hyper-casual, live-ops with teams up to 500 people. We started with classical Lean tools - Kanban, VSM, 5S - but quickly learned that game development requires two adjustments.

First, creativity is not linear. You cannot break a game designer's "mood" into equal time units.

Second, AI has changed the economics of routine work. What used to take an hour of a mid-level specialist now takes ten seconds of generation plus five minutes of verification.

Classical Lean teaches waste removal. Smart Lean teaches waste design to zero - not on paper, but inside a live digital twin of your production conveyor.

3. THE BASE UNIT: HYBRID CELL INSTEAD OF HANDOFF CHAIN

In a typical studio, an artifact (concept, level, script) passes through several specialists. Each is busy with their own work. The artifact waits. This is waiting muda - the most expensive waste category in game development.

Hybrid cell is the basic building block of Smart Lean. Inside a cell, one human specialist works alongside several AI agents. AI handles all preparatory, repetitive, and draft work. The specialist focuses only on things that require taste, lore alignment, balance decisions, and final acceptance [4].

Example from practice: A level designer builds a combat level. AI agents:

- generate four greybox enemy placement variants in two minutes based on rules from the knowledge base;
- check whether settings exceed performance limits;
- prepare the scene for the first test.

The designer does not spend two hours placing enemies manually. They spend ten minutes choosing a variant and making one adjustment based on intuition. What took three days of polishing now takes half a day.

Table 1. Traditional handoff chain vs. hybrid cell

Parameter	Traditional chain	Hybrid cell
Typical cycle time (one level)	5 - 8 days	1.5 - 2 days
Waiting time share	50 - 70%	<15%
Quality ownership	distributed ("not my stage")	single specialist (final accept)
Feedback speed	next iteration = next sprint	within the same day

4. DIGITAL TWIN OF THE FLOW: VSM THAT DOES NOT GATHER DUST

Value Stream Mapping in classical Lean is a one-time diagram. You draw it at the start, put it on the wall, and forget it. In Smart Lean, VSM becomes a live digital twin of the entire production conveyor [1].

What this delivers in practice:

- Every work item (task, asset, build) has a real-time status in the stream.
- Visibility is not limited to "in progress". The system shows "waiting for AI", "waiting for specialist review", "blocked by quality failure".

- Every AI run is logged with time cost and output result.
- A producer sees a bottleneck not during a standup meeting but by fact: an item stayed two days at ”AI texture generation” - the agent ran out of quotas or the prompt broke.

VSM as a digital twin is not a dashboard for its own sake. It is a tool that automatically highlights where waste is happening right now. No manual measurement, no guesswork.

5. THE PDCA CYCLE INSIDE SMART LEAN

Any AI agent in Smart Lean does not work as ”just do it”. It operates through a Plan → Do → Check → Act cycle. This turns AI from a noise generator into a controlled instrument.

Plan

The agent receives a machine-readable package - not ”make a beautiful texture” but ”wall texture, medieval fantasy style, 2K format, 8MB memory budget, no visible tiling, color palette №14”. The plan is a technical specification with measurable criteria [3].

Do (sandbox execution)

The agent works in an isolated environment. It can generate five to ten variants, run local checks (size, format, IP violations), and break nothing in the main build. This is critical: AI never touches production until a human says yes.

Check (predictive quality)

Before the specialist sees the result, the system compares it against the plan. For code - automated tests and static analysis. For art - style similarity against references (percentage match), budget checks. If the result fails, the agent either makes a second iteration itself or highlights the issue.

Act (learning loop)

The specialist makes the final decision: accept, reject, or edit. But the key step is recording the **delta**: what was changed, which instruction was added, which case was missed. On the next run, the agent starts with the updated rule.

From practice: After two to three weeks of such a cycle, an agent generating UI assets produces output that previously required corrections in 70% of cases - now in 20%. No AI code rewriting. Just accumulated rules.

5.1 FEEDBACK LOOPS BETWEEN HYBRID CELLS

A single hybrid cell works well for isolated tasks. But game production involves dozens of cells working in parallel or sequence. The waste reappears at the boundaries between cells unless feedback loops are explicitly designed.

In Smart Lean, each cell publishes three artifacts:

- **Output** (asset, code, build, or decision).

- **Quality certificate** (what checks passed automatically).
- **Context log** (which rules were used, what variants were rejected, why).

The downstream cell does not start from zero. It reads the context log and runs its own AI agents against the received output before a human touches it. If the upstream cell generated a texture that violates performance budgets, the downstream cell's QA agent flags it within minutes - not days.

This turns handoffs from lossy transfers into stateful relays. Every cell knows what the previous cell intended, what it tried, and what it rejected. The human specialist only steps in when the automated relay detects an exception.

Table 2. Handoff types and waste levels

Handoff type	Typical waste (waiting + rework)	Smart Lean with feedback relay
Design → Art	2 - 3 days	2 - 4 hours
Art → Integration	1 - 2 days	1 - 3 hours
Integration → QA	12 - 24 hours	30 - 90 minutes
QA → Rework loop	1 - 3 days per iteration	ame day, multiple iterations

6. KAIZEN WITHOUT MONTHLY RETROSPECTIVES

Classical Kaizen means continuous improvement through meetings. In Smart Lean, improvements are pulled from data:

- Telemetry from builds (where players get stuck or quit).
- Behavior of QA agents (which scenarios break most often).
- Production tracker data (which tasks exceed estimated time).

Instead of twenty-page reports - micro-feedback:

"On level 3, players die at the first turn in 68% of attempts. Reduce enemy health by 15% or place a medkit ten meters before that point."

The AI agent generates such hypotheses itself. Leads only approve or reject. The improvement heatmap shows where one change yields the highest expected value.

This allows issues to be closed on the day they appear - not weeks later. Critical for live-ops, where the cost of a bug or imbalance is one day of lost retention.

6.1 FROM HEATMAP TO ACTION: THE ROLE OF THE LEADS

A heatmap of problems is not enough. Someone must decide which improvement to fund and which to postpone. Smart Lean does not automate this decision - it structures it.

Every issue on the heatmap comes with three numbers:

- **Frequency** (how many players / tasks / builds were affected in the last 24 hours).
- **Expected lift** (estimated improvement in retention, cycle time, or cost if fixed).

- **Effort to fix** (estimated human time + AI compute).

Leads review this triage once per day, not once per week. They spend five to ten minutes selecting the top two or three items. AI agents then draft proposed changes (e.g., ”adjust checkpoint position”, ”rebalance enemy health”, ”add two validation rules to the texture agent”).

The lead does not write the change - they review and approve. The execution is already prepared.

This pattern is borrowed from lean manufacturing’s andon system but adapted to digital production. The heatmap is the andon cord that anyone can pull, but the response is systematic, not heroic.

7. SMART 5S: NOT DROWNING IN AI-GENERATED CLUTTER

AI in game development creates a new problem: volume. One texture generation produces hundreds of options. Dialogues are created in dozens within an hour. Builds pile up in bundles. After a month of work in the studio, no one can find the right asset. Searching becomes a guessing game.

Smart 5S solves this problem not through discipline (which doesn’t work when there are 50+ people), but through automation. Here is how the classic five (Sort, Set in order, Shine, Standardize, Sustain) looks like in the AI-loaded studio version:

- **Sort (RAG-contextualization)** - The system pulls only the references and rules relevant to the specialist’s current task. Everything else is hidden but accessible on request.
- **Set in order (AI asset management)** - AI tags assets automatically: ”this is final”, ”this is draft from agent №3”, ”this is lore document version 2.1”. Search works by meaning, not by filename.
- **Shine & Standardize (auto-linting)** - Every night, cleaning agents run: remove unused assets, recompress textures, fix naming, update documentation.

In a studio that implemented Smart 5S, asset search time dropped from 25 minutes to two minutes. Onboarding time for a new designer fell from two weeks to two days.

7.1 THE COST OF NOT HAVING SMART 5S

Studios that skip Smart 5S often discover hidden costs three to six months after introducing AI agents. The symptoms are recognizable:

- Multiple versions of the same lore document circulate, causing inconsistencies in dialogues and quests.
- Prompts are lost. The same mistake is corrected twice because no one remembers the previous fix.
- Onboarding new hires takes weeks because the knowledge base is a mess.

Smart 5S turns these costs into measurable metrics. For example:

- **Search hit rate** (how often a search returns the needed asset within three attempts).
- **Prompt reuse rate** (how many tasks reuse an existing prompt instead of writing a new one).

- **Version conflicts per week** (how many times two specialists work from different versions of the same reference).

When these metrics drop below a threshold, the cleaning agents run more aggressively or notify a lead that the knowledge base needs restructuring. The system does not wait for human complaint.

8. WHAT THIS DELIVERS IN PRACTICE: RESULTS FROM IMPLEMENTATIONS

Over twelve years, the methodology has been applied in fifteen studios to varying degrees. Average results after six to nine months:

Metric	Before	After (6 - 9 months)
Production cycle (idea → playable feature)	4 - 6 weeks	1.5 - 2 weeks
Waiting time share in cycle	50 - 70%	10 - 20%
Patch / hotfix delivery time	2 - 3 days	2 - 6 hours
Specialist time spent on routine	40 - 60%	<15%
Estimation accuracy (actual vs planned)	+120 - 150%	+30 - 40%

The key change is not speed by itself. It is that the studio stops fearing iterations. You can ship a feature, collect data, and rebuild it in two days instead of one month. For live-ops, this is the difference between a game living or dying [1].

8.1 CASE EXAMPLE: LIVE-OPS PATCH CYCLE BEFORE AND AFTER SMART LEAN

Consider a typical live-ops scenario: a game ships a new event. Within six hours, telemetry shows a 15% drop in retention at a specific boss fight.

Before Smart Lean:

- Day 1: Data analyst spots the drop, writes a report, sends to production.
- Day 2: Production meets to prioritize. Designer proposes balance change. Art needs a new visual cue.
- Day 3: Art generates assets. QA tests manually.
- Day 4: Patch built, certified, submitted.
- Day 5: Patch live. Retention loss continues for five days.

After Smart Lean (with hybrid cells and live heatmap):

- Hour 1: AI agent detects retention drop, correlates with boss fight, generates three hypotheses.
- Hour 2: Lead designer reviews hypotheses, picks one (”increase warning telegraph before boss attack”).
- Hour 3: AI agents generate two animation variants and one UI variant. Automated checks pass.
- Hour 4: Specialist approves. QA agents run 200 automated playthroughs. No regression.
- Hour 5: Patch built and deployed (or config change pushed).
- Hour 6: Telemetry shows recovery starting.

The difference is not just speed. It is that the studio can afford to experiment. If the first fix fails, they try another the same day.

9. WHERE TO START: A STEP-BY-STEP PLAN FOR STUDIOS

If you want to try Smart Lean in your team without large upfront investment, follow this plan.

Step 1. Select one narrow stream Do not rebuild the whole studio. Pick one place where waiting hurts most: asset preparation for integration, multi-platform build assembly, or localization validation.

Step 2. Draw the current VSM with manual work vs. waiting times Be honest. You will likely see that 70% of the time the artifact is just sitting.

Step 3. Design one hybrid cell Which current actions can AI handle as a draft? Which checks can be automated before the human sees the result? Which acceptance criteria can be made machine-readable?

Step 4. Run a PDCA cycle for one task type Prompt → AI execution → automated check → human acceptance → rule update. Continue for two weeks.

Step 5. Measure the change Not ”feels faster” but measurable: share of manual time, cycle time, correction rate. If better - expand. If not - adjust prompts or change the agent.

CONCLUSION

Smart Lean is not another methodology that a studio must ”implement”. It is a way to stop fighting waste manually. In a world where AI can handle routine work faster and cheaper than humans, specialist value shifts to decision-making, creative supervision, and flow management [5].

The studio that builds its conveyor as a digital twin with hybrid cells, stateful handoffs, data-driven Kaizen, and Smart 5S will gain a competitive advantage not of 10% but of two to three times in feature delivery speed and cost.

Everyone else will just generate clutter faster.

Sources

[1] Sauer, Carl René, and Peter Burggräf. ”Hybrid intelligence-systematic approach and framework to determine the level of Human-AI collaboration for production management use cases.” *Production Engineering* 19.3 (2025): 525-541.

[2] Boeira, JN Rosenfield, and J. Naomi. ”Lean Game Development.” *Apress, Berkeley, CA* (2017).

[3] Ali, Nauman Bin, Kai Petersen, and Breno Bernard Nicolau De França. ”Evaluation of simulation-assisted value stream mapping for software product development: Two industrial cases.” *Information and software technology* 68 (2015): 45-61.

[4] Shahin, Mohammad, et al. ”Improving operations through a lean AI paradigm: a view to an AI-aided lean manufacturing via versatile convolutional neural network.” *The International Journal of Advanced Manufacturing Technology* 133.11 (2024): 5343-5419.

[5] Gao, Shanghua, et al. ”Empowering biomedical discovery with AI agents.” *Cell* 187.22 (2024): 6125-6151.