SDLC Bottleneck! Game Rules

The backstory
You're consulting for MidCorp, a mid-sized enterprise with a typical product delivery pipeline. Like many established companies, they've built a system that delivers features to customers, but leadership sees room for improvement.

The current state:
MidCorp runs a Kanban-style flow where features move step by step. No heavy project management, just a continuous stream from idea to production.

Game objective

1. **Find the bottleneck**: Identify where work piles up.
2. **Select the right solution:** Apply AI at the actual constraint.(game cards)
3. **Understand the impact:** See how targeting the right (or wrong) constraint changes throughput.

One round = One day
Every round equals a working day. Tokens represent features moving through the system.

Demand scenarios
At the start of each day, new tokens are added (pick one scenario for the entirety of the game):

- Scenario A: 3 per day (high demand)
- Scenario B: 2 per day (medium demand)
- Scenario C: 1 per day (low demand)

Daily sequence

Flip tokens on each move to track progress (Each token represents a task):

1. On each day, add new tokens to the BA queue (according to the demand scenario you chose).
2. Move all active tokens one step to the right, starting with QA tasks, then Dev, and finally BA.
3. If there is no queue backlog, tokens may move directly into an empty row.
4. Empty rows can accept tokens from the input queues: QA, Dev, BA.
5. If a token reaches the ★ square cell, roll the dice before moving to check if rework is required. If rework is needed, the token remains in the same queue.
6. Proceed to the next day, and continue for the number of days you decide to play. (15-30)

You'll need about 30 tokens to play.

Rework rules (★ Squares)

Some steps are marked with a ★, meaning they carry rework risk:

- Before moving tokens into a ★ square, roll a die to simulate the % of rework.
- Example rework (30% chance):
    - Roll 1–2 → Rework: tokens return to that step's designated rework queue.
    - Roll 3–6 → Proceed: tokens move to the ★ square and continue next day.

Critical constraint

Each token occupies its entire row until it exits. Even if it's midway through, that row is blocked. New tokens can only enter once the row clears.

The teams & constraints

- **Demand:** Feature requests arrive steadily (3, 2, or 1 per day, depending on scenario).
- **BA Team:** Very efficient—can complete 3 features in a single day.
- **Engineering:** Mixed seniority, can build 5 features in 2 days. However, 30% of incoming work isn't ready and bounces back to BA.
- **QA:** Tests 2 features in 2 days. Finds defects 30% of the time, sending work back to Engineering.
- **DevOps:** Limited staff means deployments only every 5 days during scheduled windows.

Queue rules

Batch deployment

- DevOps deploys only on days 5, 10, 15, 20, 25, and 30.
- Tokens accumulate in the Deploy Queue until the next batch.
- On deployment days, all waiting tokens are released to END.

Queue behavior

- Queues have unlimited capacity.
- Tokens skip the queue if a row is empty.
- Tokens wait if all rows are blocked.

Metrics to track

Per round:

- **Throughput:** Total tokens delivered by day 30.
- **Max queue sizes:** Largest backlog at each station.
- **Bottleneck:** Where the biggest queue forms.
- **Lead time (advanced):** Write entry and exit days on tokens to calculate duration.

Key observations:

- Which queue grows the most?
- Where do tokens wait longest?
- How does demand level shift the bottleneck?
- What happens when AI is applied at the wrong steps

.

AI gamecards to apply/print

The Story

LLM Copilot helps developers better understand BA stories by clarifying requirements in real-time. AI generates code, catches bugs before commit, and suggests optimizations. When unclear requirements arise, AI helps interpret them correctly, reducing the need to send items back to BA for clarification.

Step Impact

- 50% faster (2→1 day)
- +20% capacity (5→6 rows)
- **Reduces rework: 30% → 15%**
- Handles 6 features/day max

Grid Dynamics

The Story

AI helps generate user stories from conversations and requirement drafts. It identifies uncertainties, contradictions, and suggests fixes for identified issues. BAs can focus on stakeholder alignment while AI handles routine analysis.

Step Impact

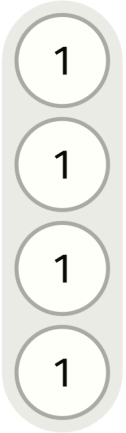
- +33% capacity (3→4)
- Same 1-day speed
- Handles 4/day
- Better quality stories for downstream

Grid Dynamics

AI Code Generation

1 day × 6 capacity **15% rework**

AI Business Analysis



1 day × 4 capacity

The Story

AI-powered test generation creates comprehensive test suites from requirements and user stories. Intelligent test execution prioritizes high-risk areas and better distinguishes real bugs from false positives. The AI provides clearer root cause analysis, reducing incorrect bug reports sent back to Development.

Step Impact

- 50% faster (2→1 day)
- 100% more capacity (2→4 rows)
- **Reduces rework: 30% → 15%**
- Handles 4 features/day max

Grid Dynamics

AI Test Automation

1 day × 4 capacity **15% rework**

The Story

AI-orchestrated deployment pipeline with intelligent rollback capabilities. Automated canary deployments, real-time monitoring, and predictive failure detection enable daily deployments instead of waiting for 5-day batch windows. Zero-downtime deployments become the norm.

Step Impact

- Daily deployments (vs every 5 days)
- Eliminates batch delays
- Instant delivery to production
- Continuous flow enabled

Grid Dynamics

AI Continuous Deployment

Deploys every day + Unlimited capacity