Game Theory for Products: Coalitions, Incidents, and Debt

Part II. Prepared by Kristina Egorova

Recap - What We learned in Part I

Setup:

- We have 4 players (PO, QA, Dev, and Manager)
- Each player has type and strategy

All work on one product

Takeaways from Part I:

- Your actions signal your type — especially escalation.
- Watch behavior to update beliefs — types are hidden.
- Escalation costs trust, even when it works.
- Trust boosts delivery — it’s part of team velocity.
- You manage incentives, not just people.

What’s New in Part II - and Why It Matters

New	Why it matters	Assumption in Part I
Coalitions emerge	Team behavior isn't just individual — players align and reinforce each other	Each player acts independently
Ignoring QA has risk	Choosing delivery speed over testing creates trust erosion and incident exposure	Cutting QA has no strategic consequence
Ignoring Dev has cost	Tech debt isn't visible short-term, but it undermines delivery over time	PO can ignore Dev input if delivery happens

Part I assumed individuals act in isolation and react to PO strategy.
Part II introduces social dynamics, system consequences, and long-term effects that shift the game.

Coalitions in Teams

Following coalitions are possible:

Player	Description	Outcome Impact
PO + QA	Compromise on delivery, protect quality	Higher trust, moderate delays
PO + Dev	Push features quickly, accept QA risk	Fast now, risk later
QA + Dev	Resist PO pressure, prioritize safety & tech debt	Long-term safety, delay now

Coalitions reflect behavior patterns, not just stated goals.

Risks and Long-Term Costs

Ignoring QA:

- +10-20% chance of production incidents
- Reduces Manager trust with each incident
- Future releases may face extra scrutiny

Ignoring Dev:

- Technical debt accumulates
- Delivery speed degrades over time
- One future delivery may fail

How does Game mechanics work, considering coalitions?

1. PO observes that **Dev prioritizes refactoring ove**r sprint goals (strategy = T).
2. Based on this, **PO updates their belief that Dev is type = Stubborn (S),** and may lean toward QA to enforce delivery discipline.
3. **This forms a PO+QA coalition**, but risks alienating Dev, accelerating technical debt and weakening future system maintainability.

Payoff Matrix: PO is willing to negotiate

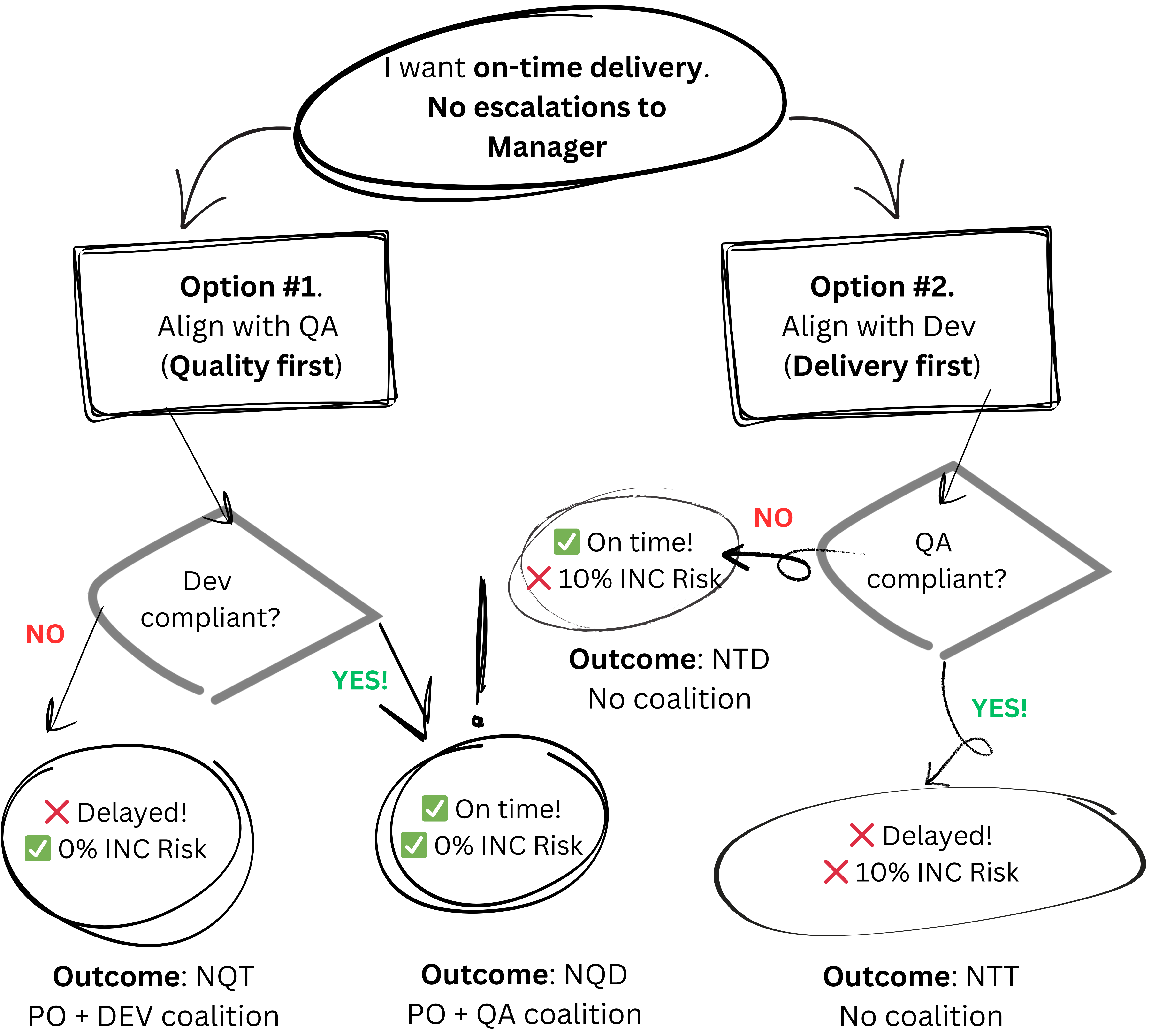
PO	QA	Dev	Delivery	Incident Risk	Trust	Coalition	Note
N	T	D	✅ On time	❌ 10%	Medium	No coalition	Delivery focused, QA compromises
N	T	T	❌ Delay	❌ 10%	Medium	No coalition	Dev focuses on tech, QA bends
N	Q	D	✅ On time	✅ 0%	High	PO + QA	Quality + delivery alignment
N	Q	T	❌ Delay	✅ 0%	Medium	QA + Dev	QA + Dev protect long-term health

You as PO are **willing to negotiate** and you are facing **time pressure** (project timelines are not-negotiable).

You don’t know the exact types of QA and Dev.

What is the **best strategy to deliver the project on time**?

Decision Tree for Product Owner



- ✅ **Strategic coalitions** are not optional — they're your real **delivery leverage**. Before decisions are made, alliances are already forming. Proactively aligning with QA or Dev defines how resilient or risky your delivery becomes.
- ✅ **Team behavior** is **shaped** by **shared incentives**, not isolated decisions. Coalitions like QA+Dev emerge to resist pressure, protect safety, or preserve maintainability — even if no one says it out loud.
- ✅ **Shortcuts aren't neutral** — they shift the system over time. Ignoring QA increases risks of production incidents and scrutiny. Ignoring Dev builds tech debt, slows future work, and reduces system stability.
- ❌ **Escalation without support weakens your position**. Even when it works today, escalation without coalition erodes trust tomorrow. It signals misalignment, isolates the PO, and reduces long-term influence.
- ✅ **Trust, alliances, and delivery move together**. You're not just managing tasks — you're managing perception and belief. The game isn't just what you ship — it's who still wants to ship with you.