Pacia
Engineering Intelligence

LEADERSHIP GUIDE

The PR lag: where software delivery quietly loses days.

How waiting between coding, review, checks, merge and QA undermines delivery confidence - and how to remove it without turning engineers into notification responders.

The leadership question

Where is elapsed time being lost after coding appears finished?

For CTOs, VPs of Engineering, Engineering Directors, Product leaders and QA leaders

01 THE SYSTEM

Busy teams create invisible queues

Developers are usually doing what the system asks: complete the story, resolve the issue and protect focused time for the next problem. A review request arrives during that work. Stopping instantly damages focus, so the PR waits.

The author moves to the next task. Then a comment arrives, approval lands, or a pipeline check fails. The state changes, but attention does not move with it. Each choice is rational locally. Together they create a queue nobody owns.

1. Code is ready

A developer opens a PR and requests review.

2. Attention moves

The author starts the next planned item.

3. State changes

Comments, approval or checks require action.

Two tasks are now in progress

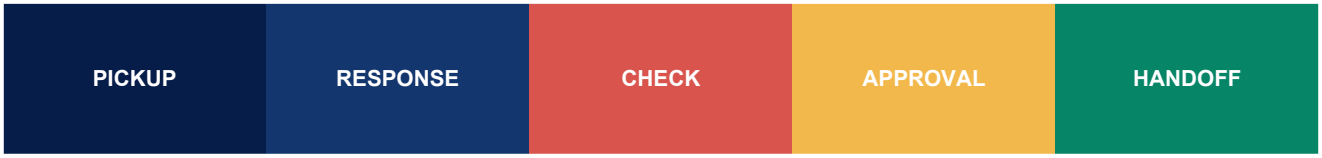
The visible task is being coded. The nearly-finished task is waiting. Delivery looks busy while completed value moves slowly.

PR lag is not evidence that developers are slow. It is evidence that waiting work has become less visible than active work.

02 THE ANATOMY

PR lag is not one wait

A single cycle-time number hides different operating problems. A stage timeline shows where the next action stopped.



Pickup lag	PR opened to meaningful review beginning	Queue visibility and reviewer availability
Response lag	Review feedback to author response	Attention has moved to other work
Check lag	Failed or pending pipeline without action	Automation stopped progress without ownership
Approval lag	Approval to merge	Release, permission or action ambiguity
Handoff lag	Code-ready to QA-planned and testable	Disconnected engineering and QA timelines

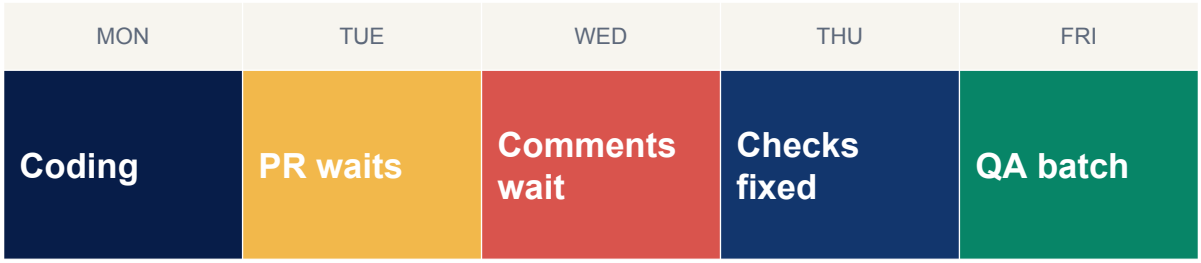
Measure the distribution, not only the average

Track time to first review, comment-to-response, time in failed checks, approval-to-merge and total PR age. Segment by team, repository, size and work type. Add reviewer concentration and queue depth. The purpose is to identify a system constraint - never to rank a person.

03 THE RIPPLE

The queue ends in QA - all at once

QA rarely receives a smooth stream of work when PR state and story state live apart. It receives uncertainty followed by a batch.



QA chases status

The issue tracker says one thing while PR reviews and checks say another.

Testing gets compressed

Late arrivals create an end-of-sprint batch and force rescheduling.

Product loses confidence

A story looks nearly complete while elapsed time remains hidden in queues.

Leadership sees surprise

The sprint slips because testable arrival was never forecast from real state.

The fix is shared operational truth: story context, PR state, review activity, checks and likely QA arrival on one timeline.

Working agreements, not individual targets

A working agreement is a team-owned expectation for how work should move. The threshold belongs to the team and its context; the value is shared expectation, not arbitrary compliance.

1. SEE THE QUEUE

Expose stage-level waiting and the work behind it.

2. CHOOSE ONE CONSTRAINT

Pickup, size, failed checks, concentration or QA handoff.

3. AGREE THE RESPONSE

Define what timely movement means for this team.

4. WARN BEFORE

Make a warning useful before the agreement is missed.

5. ROUTE THE ACTION

Send it to the smallest relevant audience in VS Code, Teams or Slack.

6. LEARN TOGETHER

Review repeated constraints in retrospectives, without blame.

The anti-pattern	The operating habit
A universal SLA imposed from above	A threshold chosen for the team and work type
Every alert sent to everyone	The next action routed to the relevant person or channel
Individual league tables	Stage trends, queue conditions and work-level evidence
Escalation after surprise	A warning early enough to protect the plan

05 THE CONNECTED WORKFLOW

Put the signal where action happens

Pacia links planned work, development state and QA readiness, then returns the relevant signal to the tools teams already use.

VS Code

PR activity and pickup-agreement notifications, with an action to open the PR.

Teams / Slack

Focused warnings and escalations for the relevant team or channel.

QA queue

Likely arrivals and confidence based on linked work and PR state.

From status chasing to operational questions

Instead of	Ask
Is it nearly done?	What state is it in, and what is it waiting for?
Can QA take this today?	When is it likely to become testable, and with what confidence?
Why did the sprint slip?	Which waiting stage became the constraint, and when did it become visible?

Find the waiting before it becomes a delivery surprise.

See PR flow, connect it to planned work and QA, and build team-owned working agreements around the real constraint.

pacia.io

06 SOURCES

Evidence and further reading

This guide combines Pacia's product model with established engineering guidance and empirical research. It deliberately avoids universal performance claims and fixed thresholds.

Google Engineering Practices - Speed of Code Reviews

Explains why team velocity depends on prompt responses while focused coding should not be interrupted. google.github.io/eng-practices/review/reviewer/speed.html

Bacchelli and Bird - Expectations, Outcomes, and Challenges of Modern Code Review

Microsoft Research / ICSE 2013. Observational, interview and survey evidence on the benefits and challenges of tool-based code review. microsoft.com/en-us/research/publication/expectations-outcomes-and-challenges-of-modern-code-review/

Czerwonka and Greiler - Code Reviews Do Not Find Bugs

Microsoft Research / IEEE 2015. Describes review as often the longest part of code integration and argues for more precise review workflows. microsoft.com/en-us/research/publication/code-reviews-do-not-find-bugs-how-the-current-code-review-best-practice-slows-us-down/

Pacia
Engineering Intelligence

Engineering intelligence for delivery flow, quality, investment, AI impact and team experience.

Visit pacia.io to see how your delivery timeline connects from planned work to QA.