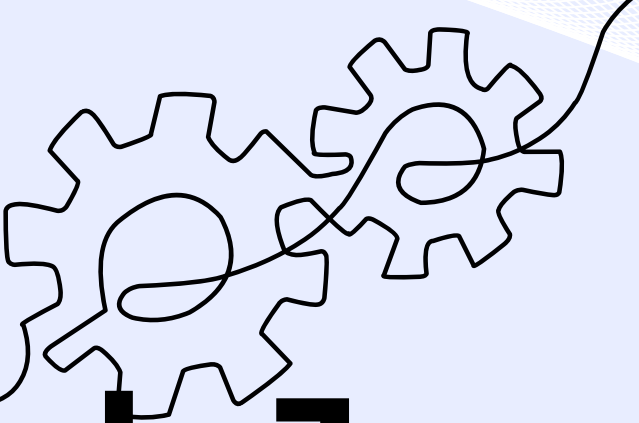




Tech Jargon^{part 1}

for
Product Managers

From Idea to Shipped

SLDC | CI/CD | DEVOPS | Deployment | Release



From an Engineer-Turned-PM

The SDLC & Deployment Mental Model

A way of seeing software work as a series of stages—from idea, to design and build, to testing, deployment, and maintenance—plus the practices and safety mechanisms that make shipping fast and reliable.

It helps PMs understand where work really is, who needs to be involved, why “just a small change” can ripple through the entire pipeline, and how modern teams ship continuously while staying safe.

When you understand this mental model, you can:

- Know which phase you’re in, so you can plan better.
- Set realistic timelines and push back on unrealistic asks.
- Understand why deployment $\neq$ launch (feature flags and dark releases).
- Coordinate releases across engineering, marketing, and support.
- Design for safety (staging, rollback, regression testing).

This guide covers 10 essential terms that unlock this way of thinking



Ar
Solonki
Arizon

SDLC

Software Development Life Cycle

The process or framework teams follow to design, build, test, and deploy software.

Think of it as the roadmap from idea → shipped product.

Engineers rarely say “SDLC” out loud—but if you hear “in development” or “ready for testing,” they’re referring to SDLC phases. These phases often overlap rather than happening strictly one after another.

Planning → Design → Development → Testing → Deployment → Maintenance

We’re still in the design phase, so locking the scope now will help us avoid expensive changes during development.



For PMs, this means you know which SDLC phase you’re in, so you can plan better, manage stakeholders, and push back on unrealistic timelines.

CI/CD

Continuous Integration / Continuous Delivery

A set of practices that automate the testing and deployment of code, so new changes can ship faster and more reliably—with automated tests catching bugs early and safe deployment steps ensuring faster releases.

Think of it as an assembly line for safe, frequent releases.

*The build broke in CI,
so we need to fix that test before merging.*



For PMs, this means your team can ship smaller, safer changes more often—but bugs can reach production faster too—so you must stay close to testing, rollout, and rollback plans.

DevOps

Development + Operations collaboration

A culture and set of practices that bring development and operations together to automate how code is built, tested, deployed, and monitored, so teams can ship changes faster and more reliably.

DevOps isn't a person—it's a system of tools and workflows that make delivery smoother.

*DevOps will handle the infra setup
we just need to define the config.*



For PMs, this means investments in automation, monitoring, and infra (infrastructure) aren't "nice to have"—they're what make faster, safer releases possible and reduce the "not my job" gaps between teams.

Deployment

Moving code to production

The act of moving code to the live/production environment. The code is running on production servers, but may not be visible to users yet (for example, if it's behind a feature flag).

This feature is deployed, but the flag is still off.



For PMs, this means deployment $\neq$ launch. Code can be live in production but hidden behind feature flags, so you can ship early and choose the right moment to launch publicly.

Staging

Pre-production environment for testing

An environment that mirrors production—same configuration and integrations where possible—used for final checks before going live.

We'll test it in staging before the release tomorrow.



For PMs, this means staging is your last line of defense. Insist that key flows and acceptance criteria are verified there before you commit to release dates with stakeholders.

Rollback

Reverting a deployment if something breaks

A safety mechanism to undo a release and restore the previous stable version in production. Rollbacks are easiest when changes are small and reversible; big database or data changes can make them harder.

We had to roll back the release due to a login bug.



For PMs, this means rollback is your safety net. Understanding how and when the team can roll back helps you decide whether to pause, continue, or adjust a launch and communicate clearly when things go wrong.

Release

Coordinated shipment of features to users

A release combines the technical deployment with the communication and activation plan so users actually adopt the change.

Let's sync on the release plan—marketing needs a heads-up.



For PMs, this means you're the bridge between "code is live" and "users know it's live." You orchestrate the release—coordination, communication, and making sure the right users see the right changes at the right time.

Hotfix

Urgent fix deployed outside the normal cycle

A quick patch to fix a critical production bug—often outside the normal sprint or QA cycle.

We'll push a hotfix to unblock signups.



For PMs, this means hotfixes are risky but sometimes necessary. You need to quickly align on impact, approve trade-offs, and communicate clearly with customers and internal stakeholders until things are stable again.

Regression Test

Testing that old features still work after
new code is added

Automated or manual tests that ensure updates
don't break existing functionality.

Let's add a regression test for that checkout bug.



For PMs, this means regression coverage is what
keeps fast shipping from turning into constant
firefighting. When you ask for speed, you should also
ask how regression will be protected.

Feature Flag

Toggling features on/off without redeploying

Feature flags let you control who sees a feature in real time, often for testing, gradual rollouts, or emergency shut-offs.

It's deployed, but we'll keep the flag off until Friday.



For PMs, this means you can de-risk launches and run experiments. You can coordinate go-live and rollbacks using flags, without always waiting on engineers to redeploy.

Stay Tuned

Up Next



A plain-English guide to how systems talk to each other — APIs, REST, GraphQL, and why it matters for PMs.

