Root Cause Analysis

Top 5 Causes of Deployment Failures

- **Inconsistent Environment Configuration:** Different setups across environments lead to unpredictable behavior, causing environment-specific deployment failures.
- **Lack of Automated Testing:** Without automated unit, integration, and smoke tests, issues go undetected until later stages, contributing heavily to failures.
- **Manual Security Scanning:** Performing security scans manually after deployment increases delays and risks, often leading to blocked or failed releases.
- **Database Migration Failures:** Improperly managed migrations cause schema mismatches and potential data loss, frequently breaking deployments.
- **Undocumented Service Dependencies:** Microservices have implicit dependencies; deploying them independently causes cascade failures across services.

Contribution to 70% Failure Rate

1. Inconsistent environment configuration: ~25% of failures due to mismatched settings and missing secrets.
2. Lack of automated testing: ~20% of failures as regressions slip through.
3. Manual security scanning: ~10% of failures caused by late security blocks.
4. Database migration failures: ~30% of failures due to schema/data issues.
5. Undocumented dependencies: ~15% of failures from cascade effects.

Prioritization & Timeline

Priority 1 (Week 1-2): Standardize environment configurations using Infrastructure as Code.
Priority 2 (Week 1-3): Automate database migrations with rollback strategies.
Priority 3 (Week 2-4): Introduce automated testing (unit + smoke tests).
Priority 4 (Week 3-4): Document service dependencies and enforce coordinated deployments.
Priority 5 (Ongoing): Automate security scanning pre-deployment.