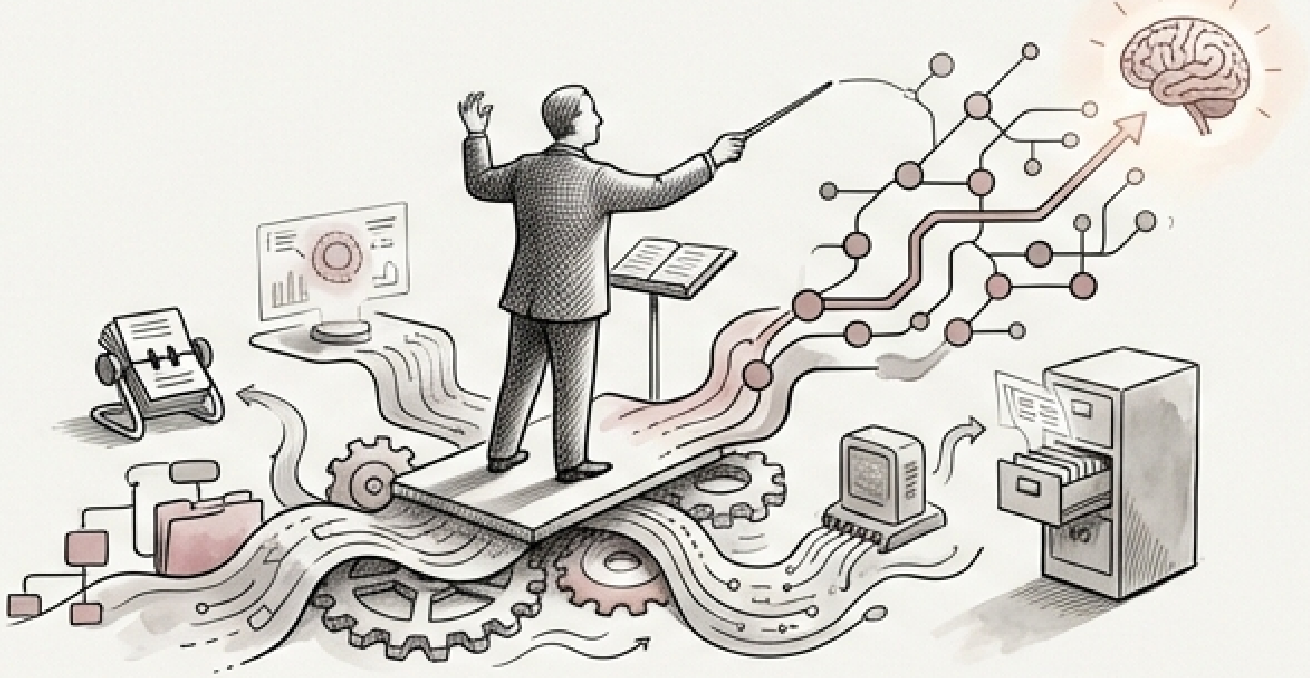
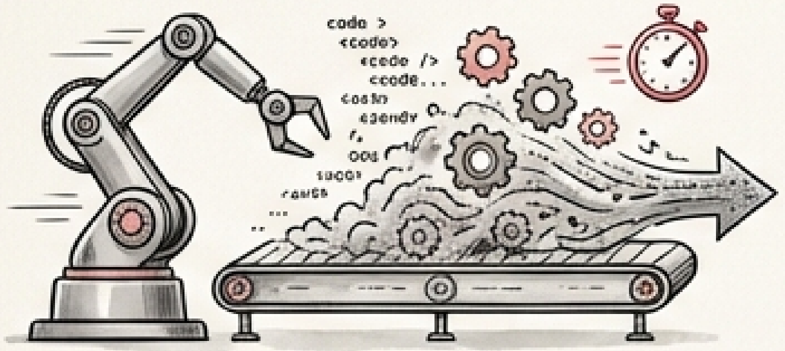
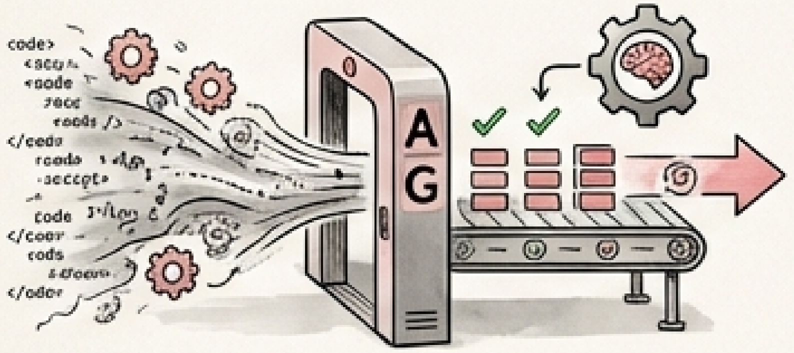
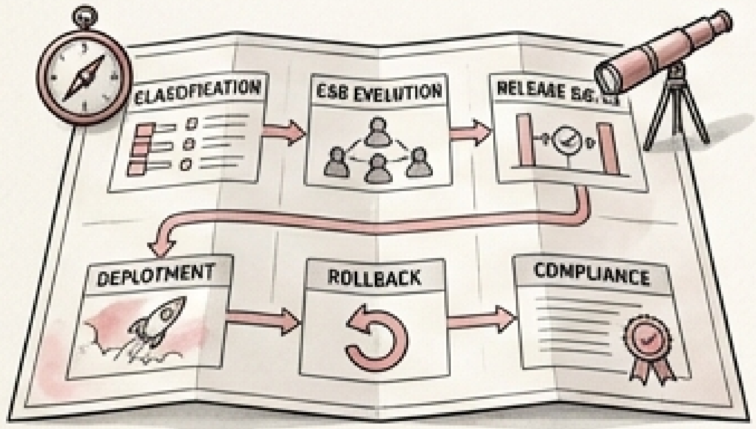


Modernizing Change Management for AI-Augmented Development

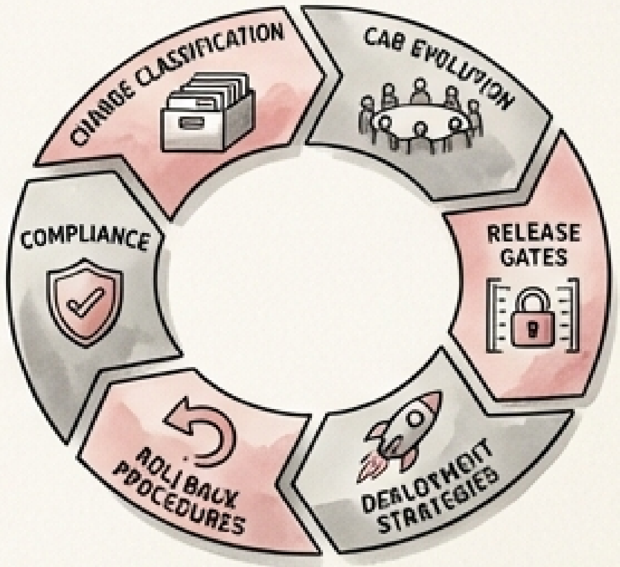
Modernizing Change Management for AI-Augmented Development





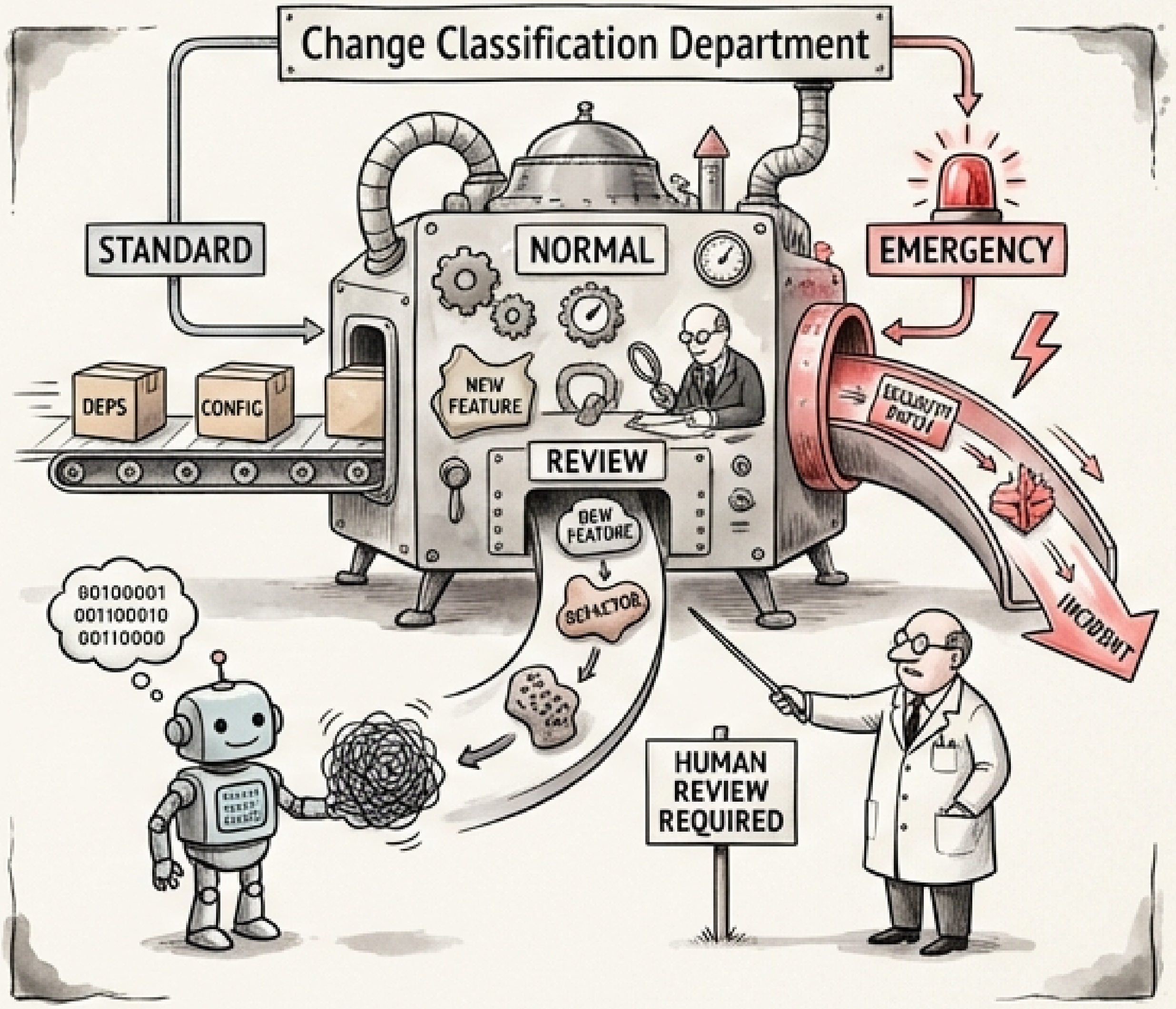


- AI-augmented development dramatically increases the volume and velocity of code changes.
- Traditional **manual change management** processes become **bottlenecks** in fast-paced AI projects.
- **Automated governance** is essential for managing the increased change frequency effectively.
- This presentation outlines strategies for **governing releases** in AI-augmented development environments.
- We will explore **change classification, CAB evolution**, release gates, deployment strategies, **rollback procedures** and **compliance**.



Change Classification: Balancing Speed and Safety

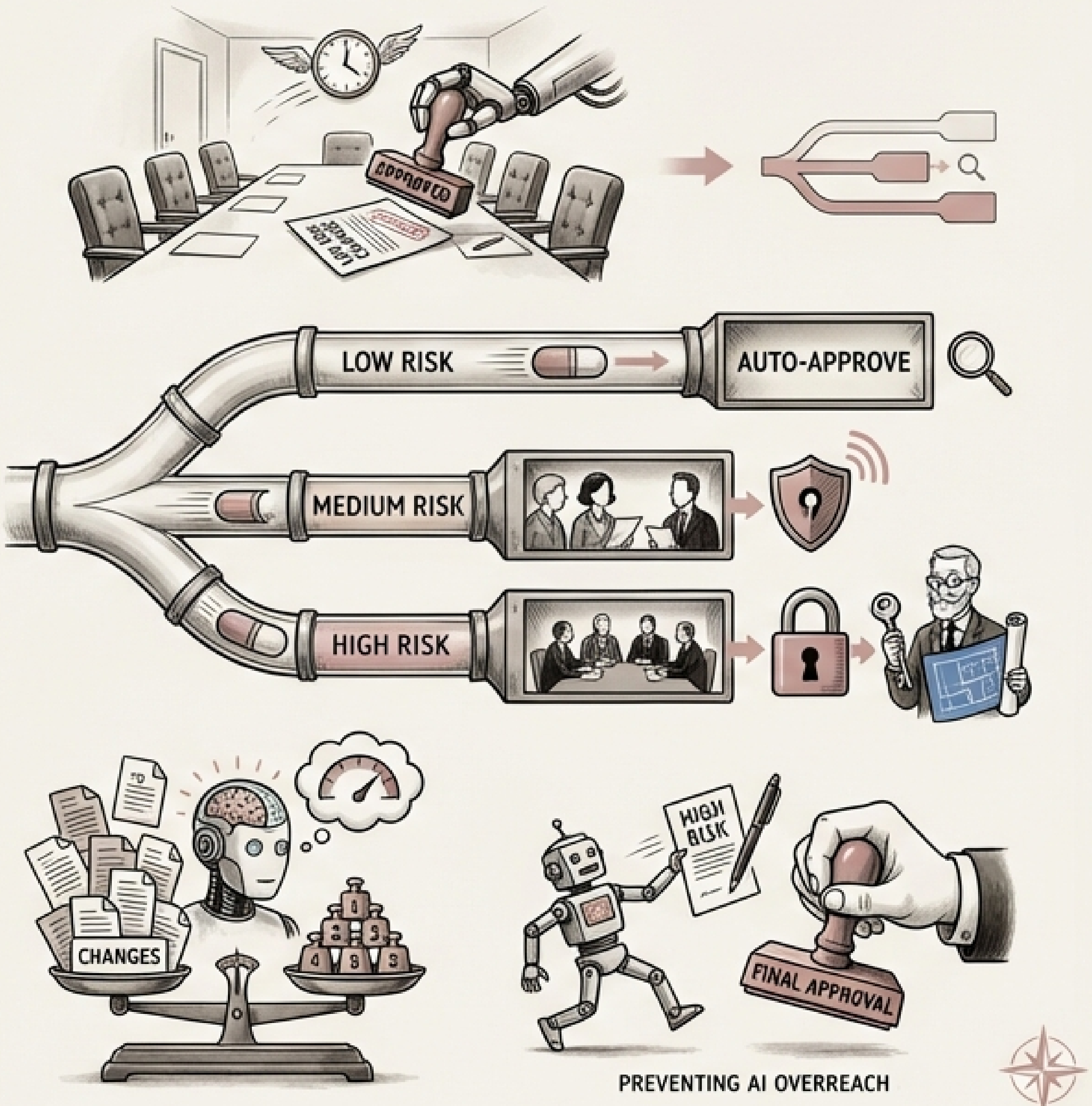
- **STANDARD CHANGES:** Pre-approved, low-risk, routine tasks like dependency updates and configuration changes.
- **NORMAL CHANGES:** Require review and approval due to moderate risk, encompassing new features and refactoring.
- **EMERGENCY CHANGES:** Expedited approval for high-urgency issues such as security patches and incident response.
- **AI-GENERATED CHANGES:** Automatically classified as normal changes, requiring human review before deployment.
- **RATIONALE:** Avoid auto-approving AI-produced code as standard changes to prevent unexpected issues.



The Balancing Act: Speed, Safety, and the Human Element in Change Management

Evolving the CAB: **Lightweight Approval Through Risk Scoring**

- Lightweight CAB: Automated risk scoring replaces manual meetings for low-risk changes, increasing efficiency.
- Risk-Based Routing: Automates workflow based on risk: low risk → auto-approve with audit trail.
- Risk-Based Routing: medium risk → peer review + security scan; high risk → CAB review + security architect approval.
- AI-Generated Risk Assessments: Triage changes and provide initial risk scores for faster decision-making.
- Human Oversight: Humans retain final approval authority, especially on high-risk items, preventing AI overreach.



Release Gates: Ensuring Quality at Every Stage

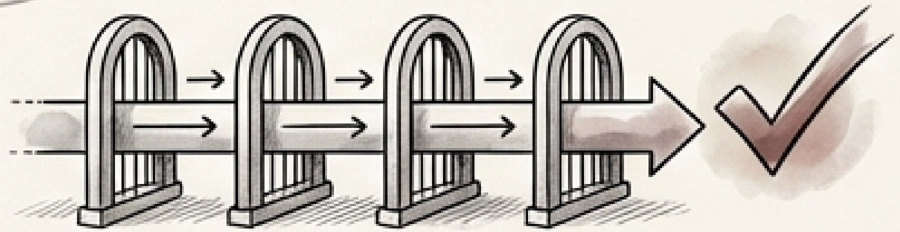
Gate 1 (Code): All tests pass, SAST clean (no critical/high vulnerabilities), SCA clean, code review approved, AI provenance tagged.

Gate 2 (Build): Artifact integrity verified, SBOM generated, container scan clean, dependency audit clean.

Gate 3 (Deploy): Environment configuration validated, rollback plan documented, monitoring alerts configured, change ticket approved.

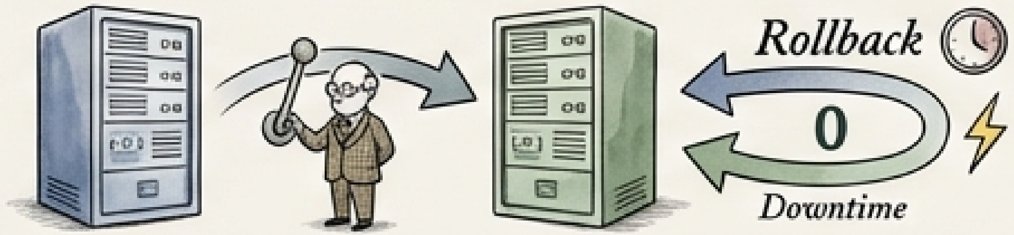
Gate 4 (Post-deploy): Smoke tests pass, error rates normal, performance baseline met, security monitoring active.

Each gate acts as a checkpoint to verify quality and security throughout the CI/CD pipeline.



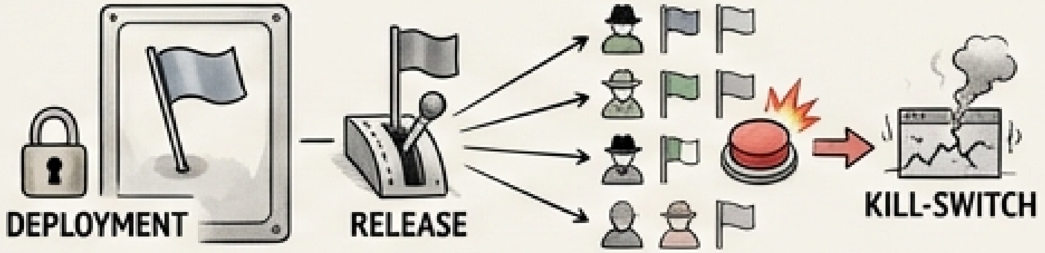
Deployment Strategies: Minimize Downtime and Risk

- **Blue-Green Deployment:** Provides instant rollback capability and zero-downtime switching between environments.

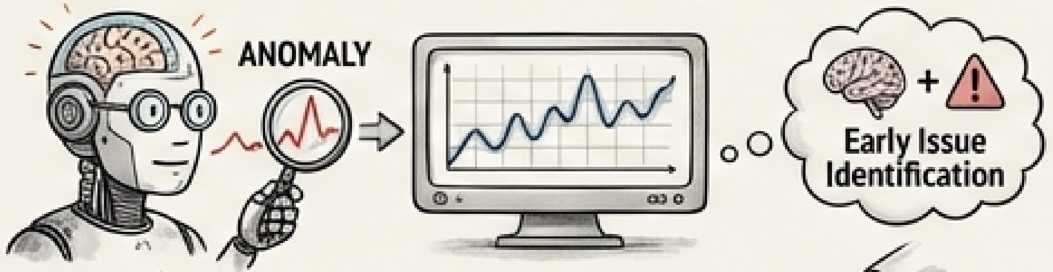


- **Canary Releases:** Gradual rollout (1% → 10% → 50% → 100%) allowing observation of impact on a smaller user base.

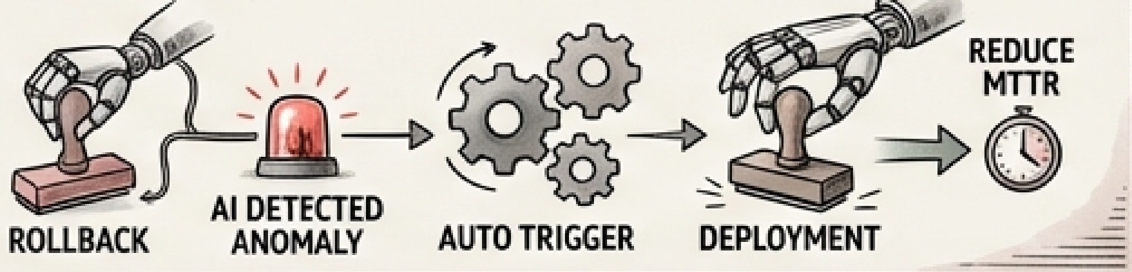
- **Feature Flags:** Decouple deployment from release, enabling targeted rollout and a kill-switch for problematic features.



- **AI-Assisted Deployment Analysis:** Uses anomaly detection on metrics during canary releases to identify issues early.

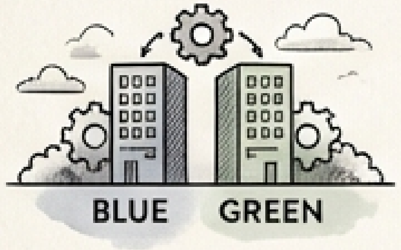


- **Automated Rollback Triggers:** Initiate rollback automatically based on AI-detected anomalies, reducing MTTR.

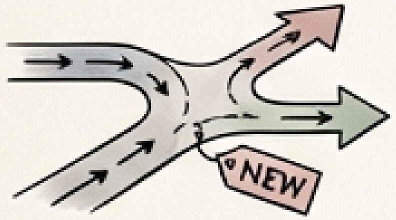


Deep Dive: Blue-Green Deployments

BLUE (LIVE)
NEW VERSION
GREEN (STAGING)



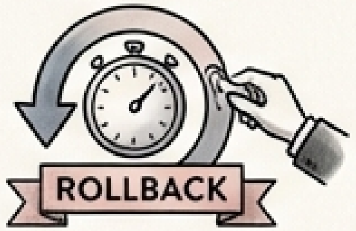
- Operate two identical production environments: Blue (live) and Green (staging).



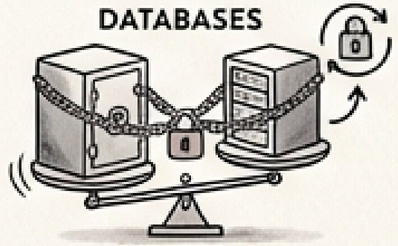
- Deploy the new version to the Green environment while Blue continues to serve traffic.

- Thoroughly test the Green environment before switching traffic from Blue to Green.

- Thoroughly test the Green environment before switching traffic from Blue to Green.



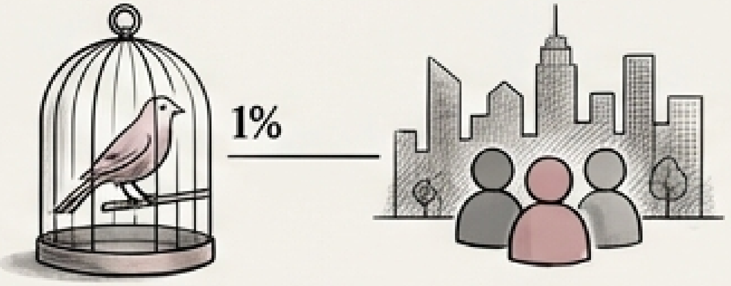
- Instant rollback capability by simply switching traffic back to the Blue environment.



- Requires careful management of databases and persistent storage to ensure data consistency during the switch.

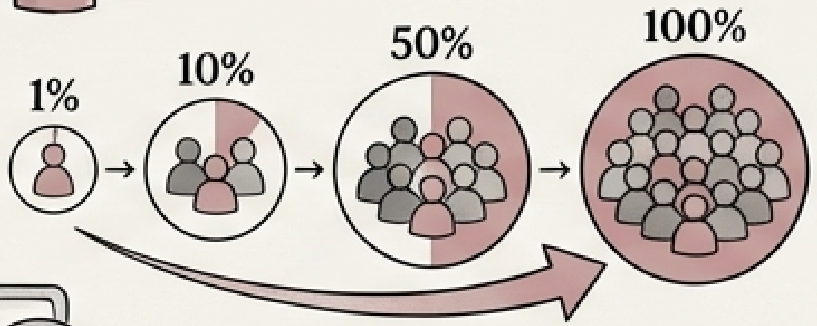
Deep Dive: Canary Releases

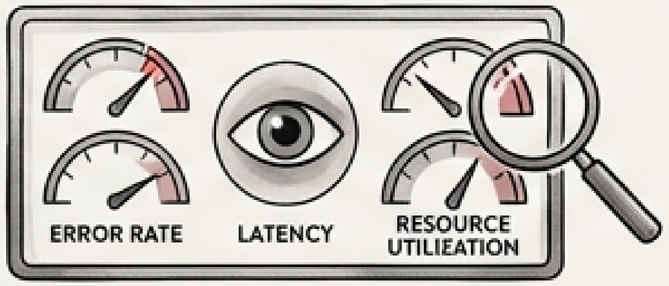
- Rollout new features to a small subset of users (e.g., 1%) to test in a real-world environment.
- Gradually increase the percentage of users exposed to the new feature (e.g., 10% → 100%).
- Monitor key metrics (error rate, latency, resource utilization) for anomalies during each stage of the rollout.
- Automated rollback triggers can be configured to automatically revert to the previous version if anomalies are detected.
- Allows for early detection of issues that may not be apparent in testing environments.



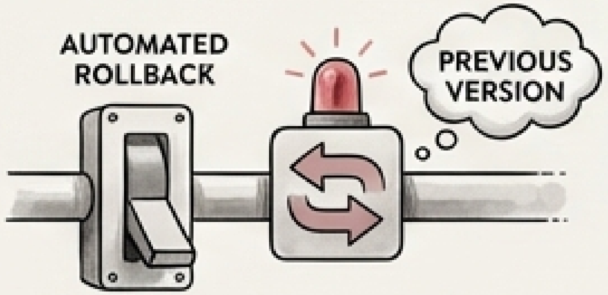
Rollout new features to a small subset of users (e.g., 1%) to test in a real-world environment.

Gradually increase the percentage of users exposed to the new feature (e.g., 1% → 10% → 50% → 100%).

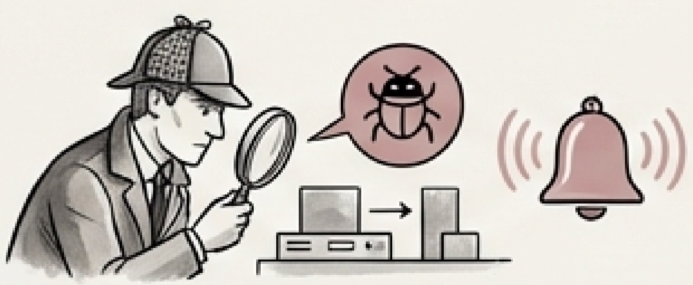




Monitor key metrics (error rate, latency, resource utillzation) for anomalies during each stage of the rollout.

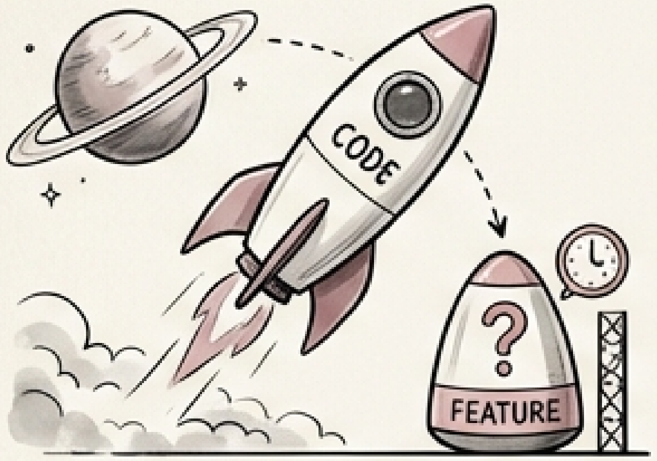


Automated rollback triggers can be configured to automatically revert to the previous version if anomalies are detected.



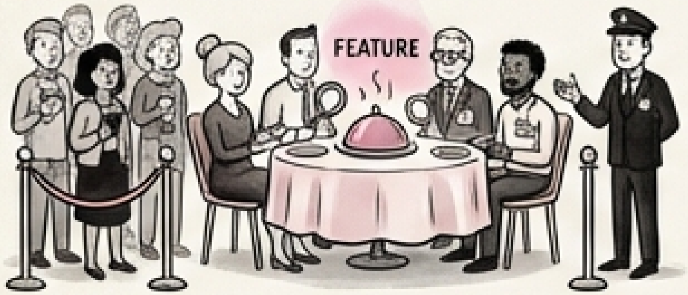
Allows for early detection of issues that may not be apparent in testing environments.

Deep Dive: Feature Flags

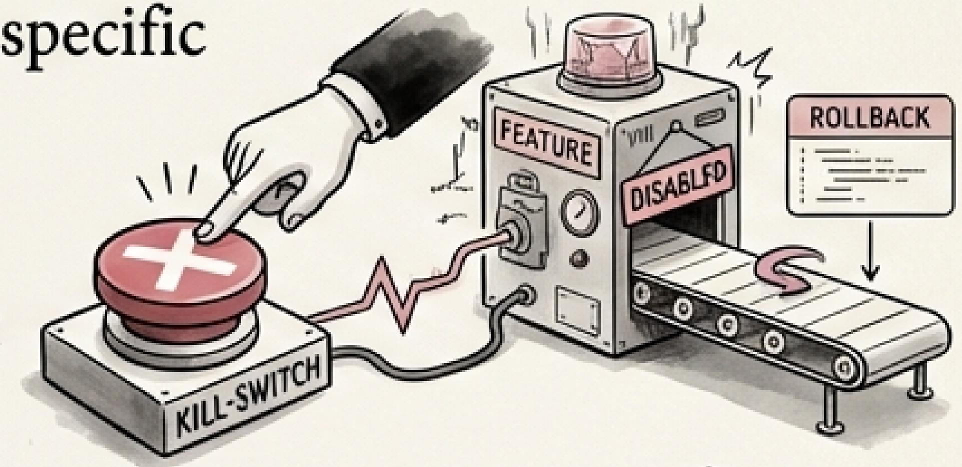


- Separate code deployment from feature release, allowing code to be deployed before a feature is ready for public use.

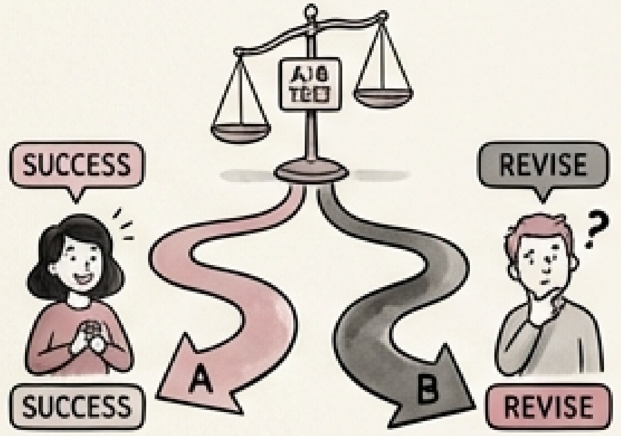
- Enable targeted rollout of features to specific user segments or beta testers.



- Provide a kill-switch to instantly disable a problematic feature without requiring a code rollback.



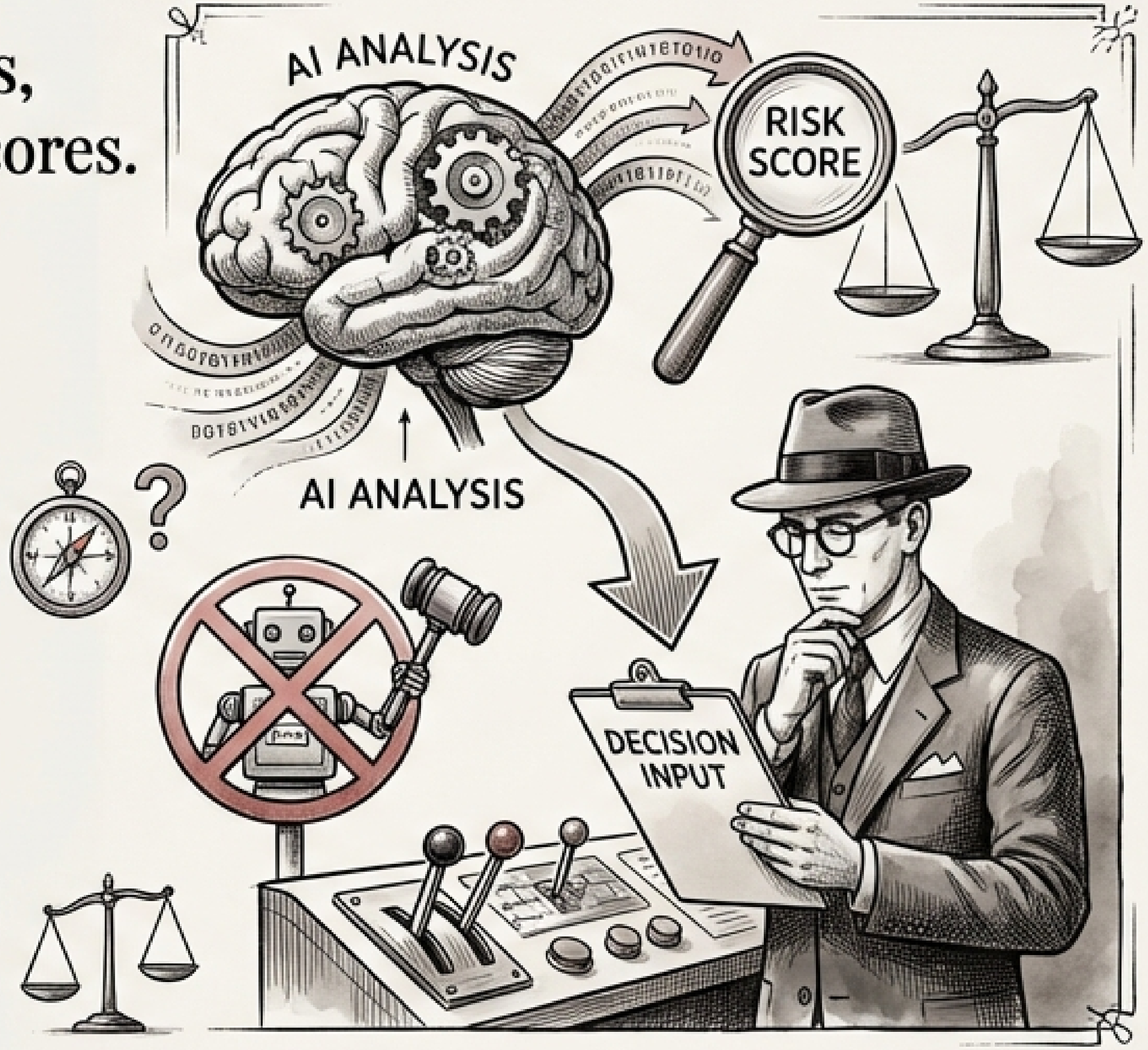
- Can be used for A/B testing to compare the performance of different feature implementations.



- Require careful management to avoid technical debt and complexity.

AI-Generated Change Risk Assessment: Augmenting, Not Replacing

- AI analyzes change scope, affected components, and historical defect data to assign initial risk scores.
- AI assists triage of changes, enabling faster identification of high-risk modifications.
- Limitations: AI lacks business context and an understanding of organizational impact.
- AI may underestimate cascading effects and the potential for unforeseen consequences.
- Crucially, AI output serves as input to human decision-making, not a replacement for it.



Rollback and Recovery: Preparing for the Inevitable

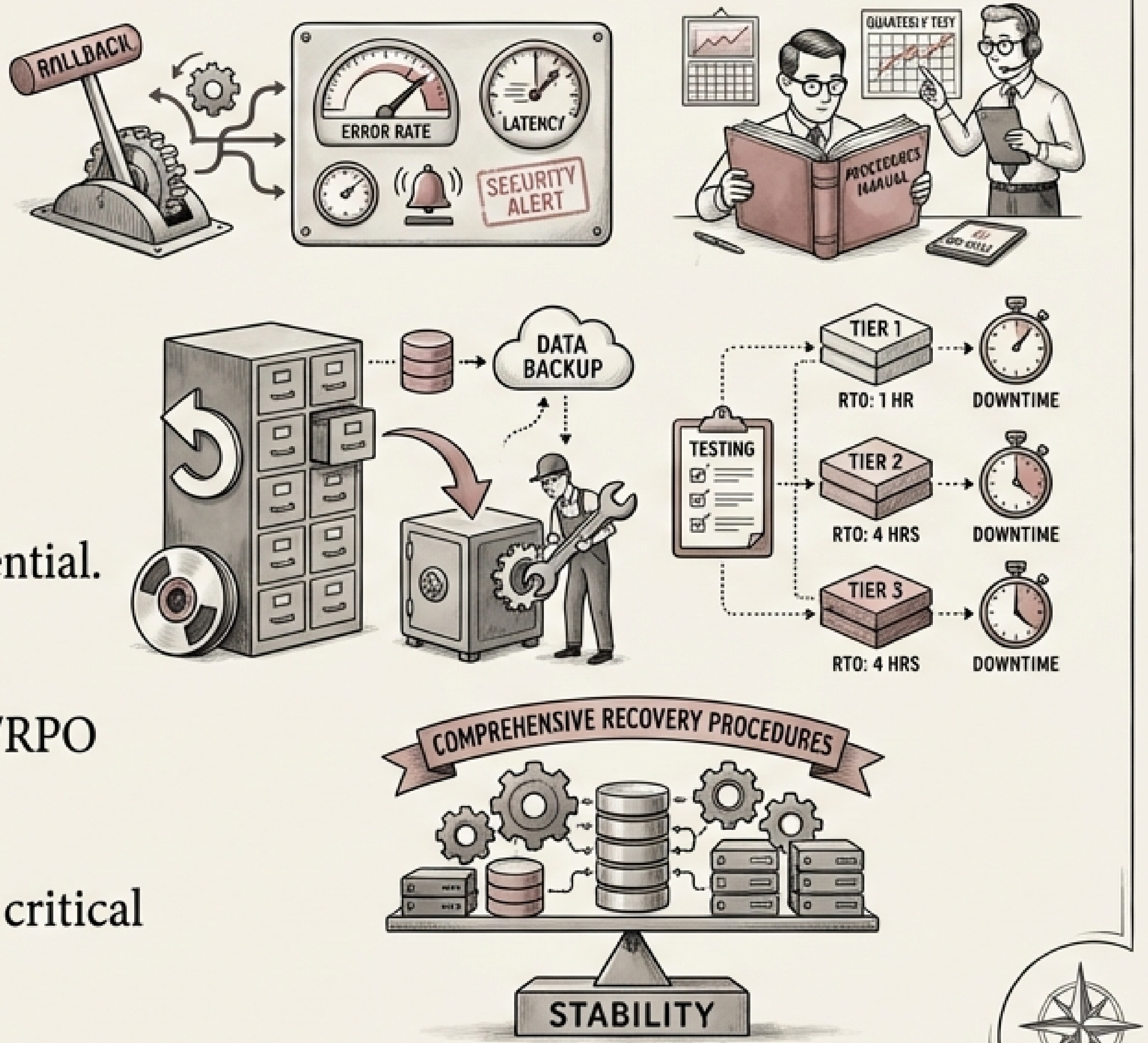
- **Automated Rollback Triggers:** Error rate exceeds a predefined threshold, Latency SLA is breached, or a security alert is fired.

- **Rollback Procedures:** Must be documented, tested quarterly, and easily accessible to on-call engineers.

- **Database Rollback:** Migration reversibility testing is essential. Always **back up data** before schema changes.

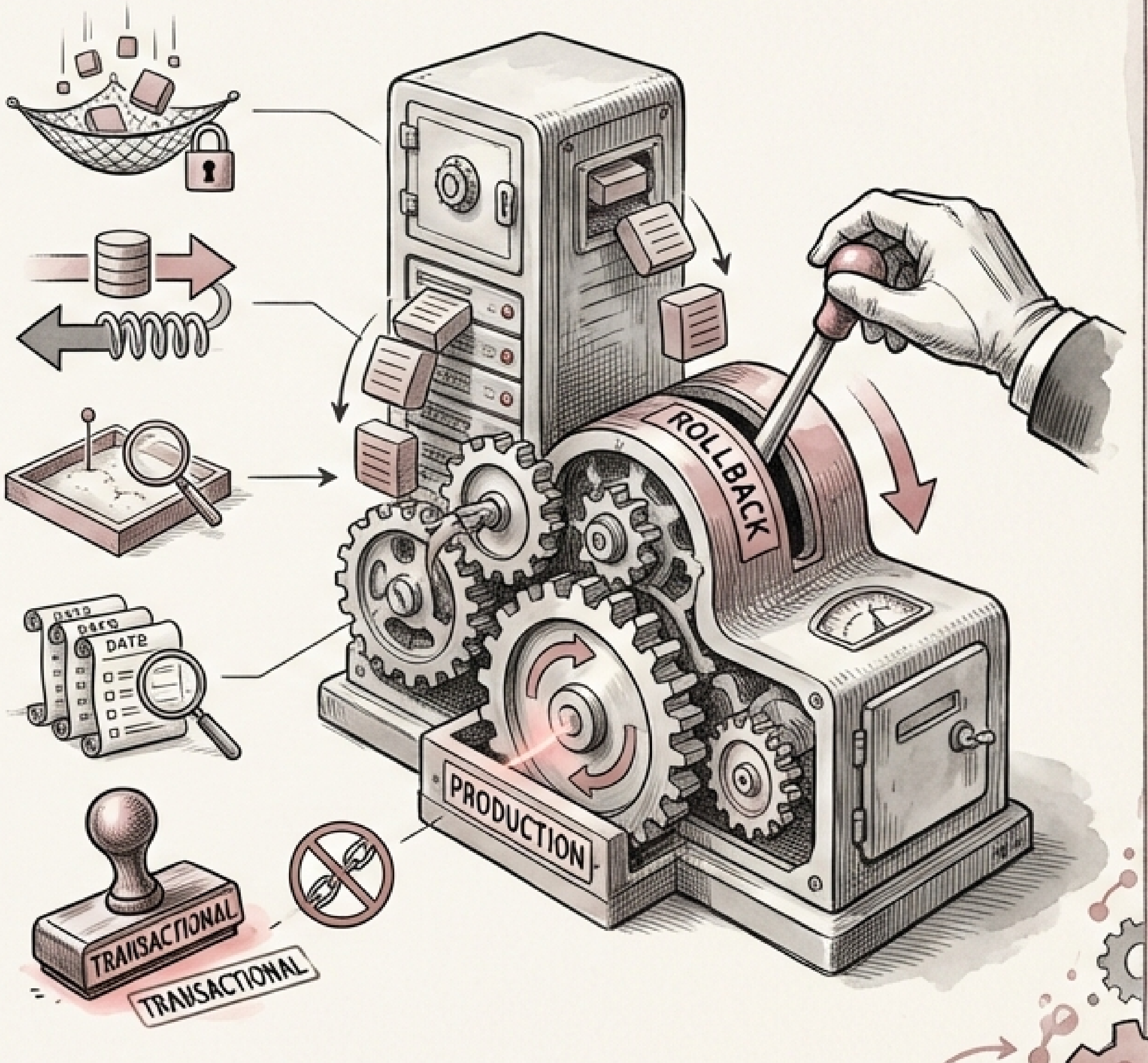
- **Recovery Time Objectives (RTO):** Define and test RTO/RPO for every service tier to minimize downtime.

- **Comprehensive rollback and recovery procedures** are critical for maintaining system stability.



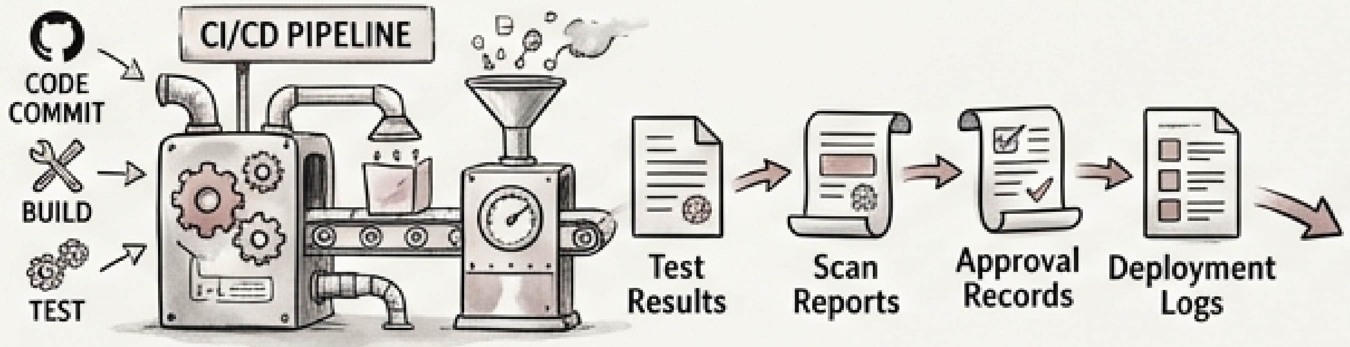
Database Rollback Strategies: Ensuring Data Integrity

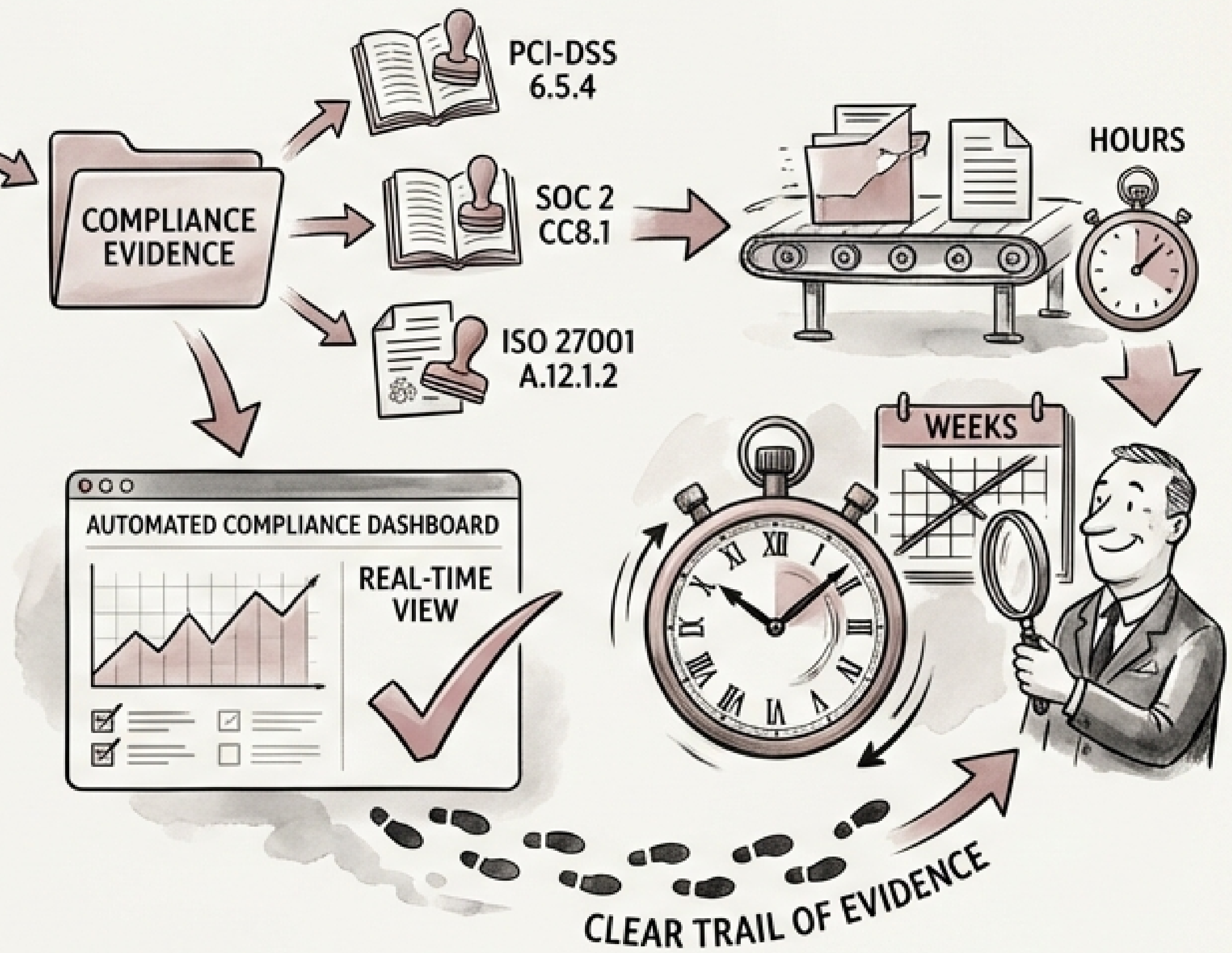
- Always create a full database backup before applying any schema changes or data migrations.
- Design database migrations to be reversible, allowing for easy rollback to the previous state.
- Test migration reversibility in a staging environment before applying changes to production.
- Implement data versioning to track changes and facilitate data recovery.
- Consider using transactional DDL to ensure that schema changes are applied atomically and can be rolled back if necessary.



Database Rollback

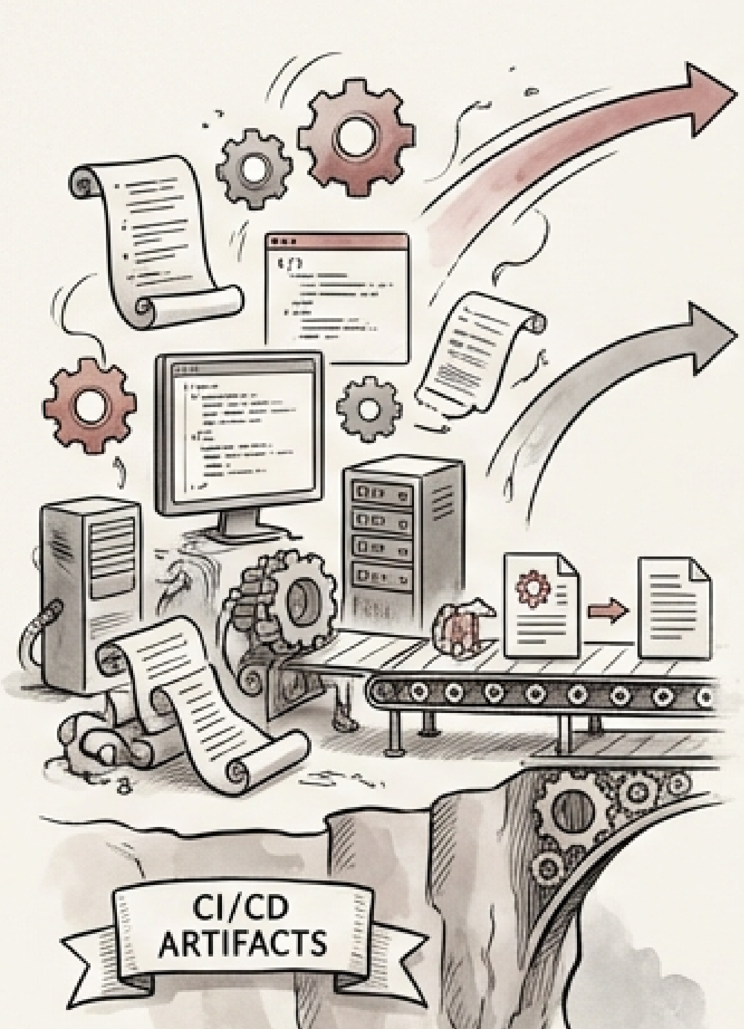
Compliance Evidence from CI/CD: Automating Audit Readiness



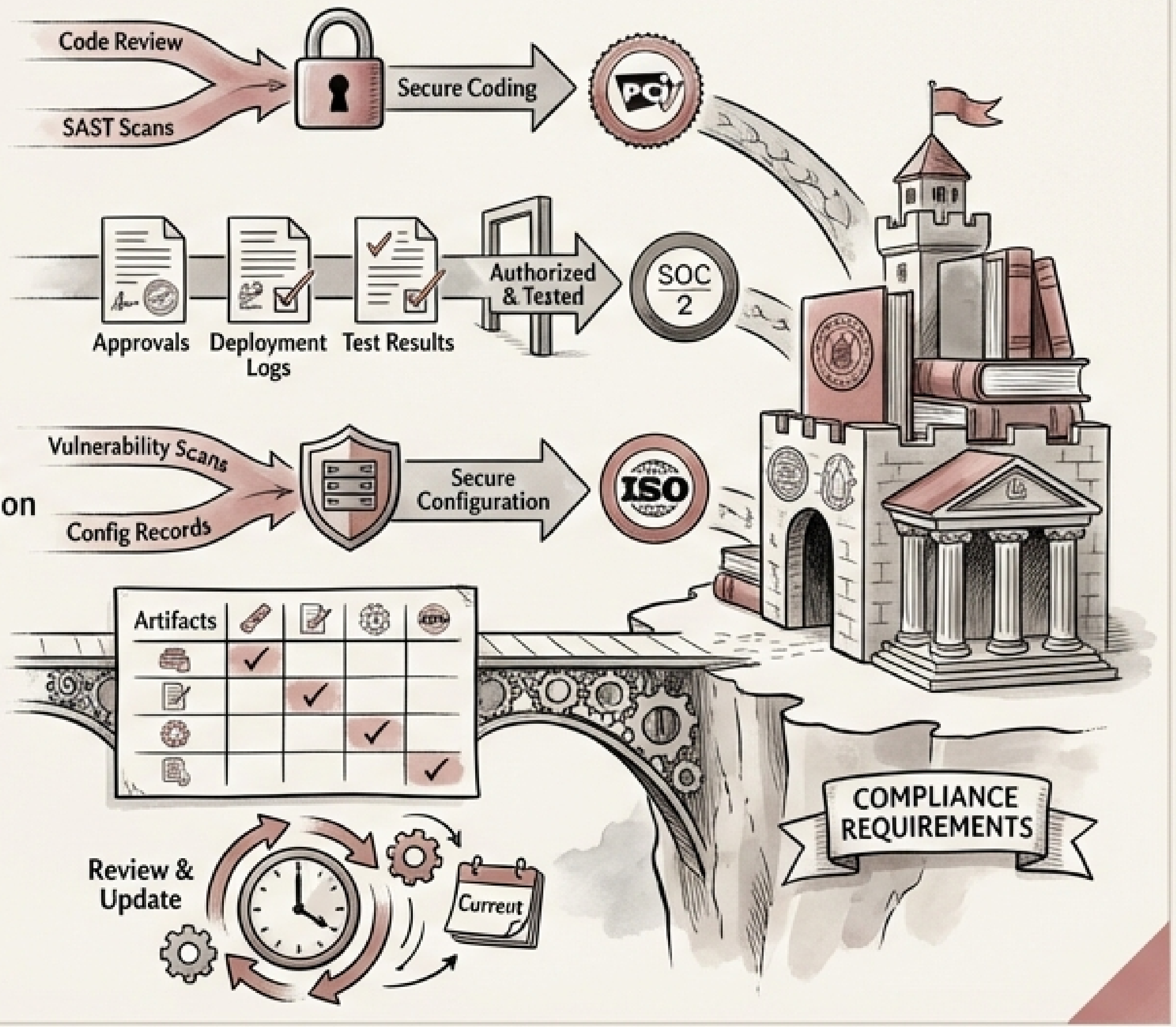


- ✓ Every pipeline run generates **compliance artifacts:** test results, scan reports, approval records, and deployment logs.
- ✓ Map these artifacts to **specific regulatory requirements** (e.g., PCI-DSS 6.5.4, SOC 2 CC8.1, ISO 27001 A.12.1.2).
- ✓ **Automated compliance dashboard** aggregates artifacts and provides a real-time view of compliance status.
- ✓ Automated compliance **reduces audit preparation time** from weeks to hours.
- ✓ **Ensure that the pipeline is auditable** and provides a **clear trail of evidence** for all changes.

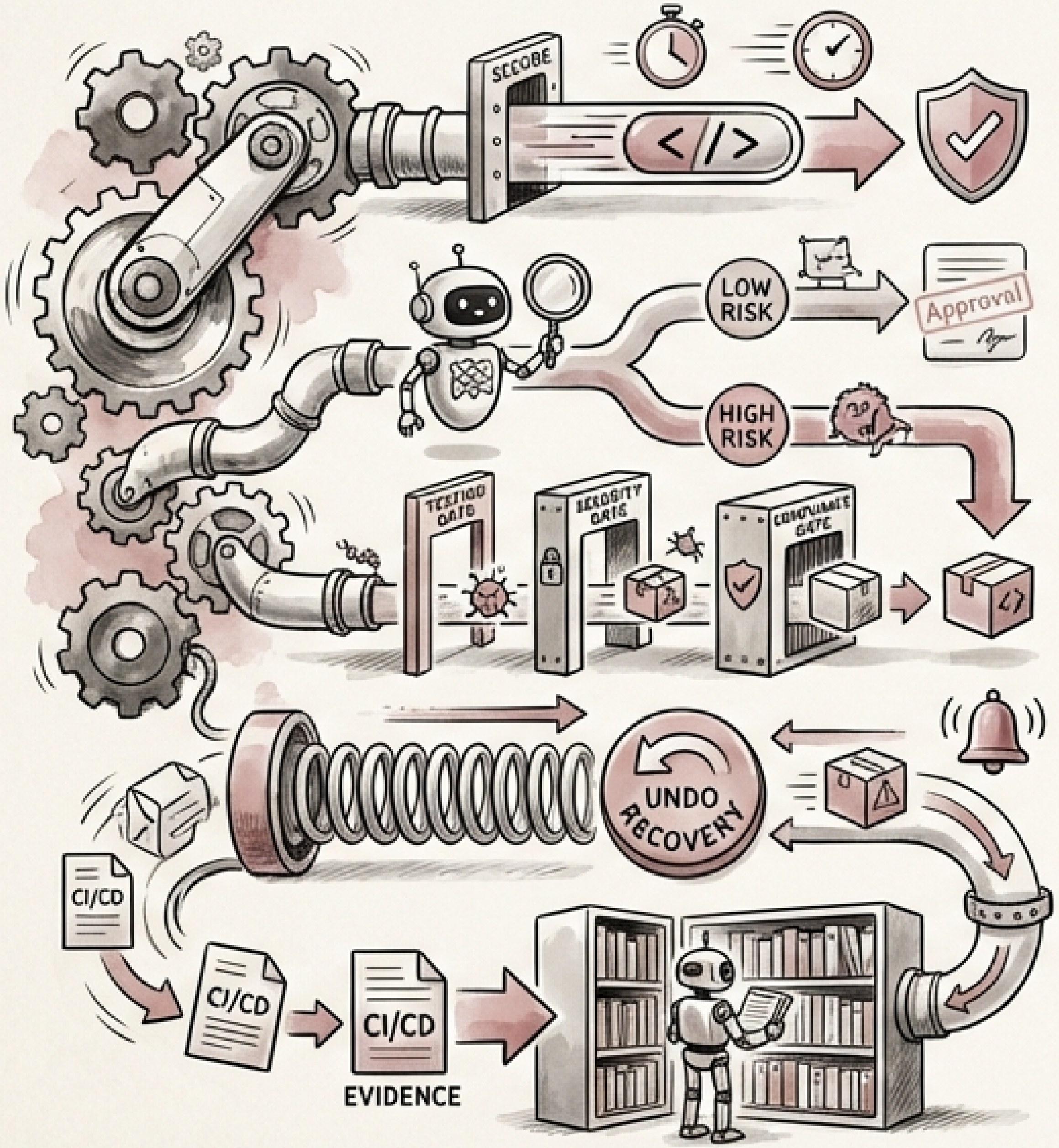
Bridging the Gap: Mapping Artifacts to Compliance Requirements



- **PCI-DSS 6.5.4 (Change Management):** Map code review records and SAST scan results to demonstrate secure coding practices.
- **SOC 2 CC8.1 (Change Management):** Use approval records, deployment logs, and test results to show that changes are properly authorized and tested.
- **ISO 27001 A.12.1.2 (Change Management):** Use vulnerability scan reports and configuration management records to demonstrate secure configuration management.
- Maintain a matrix that explicitly links CI/CD artifacts to relevant compliance requirements.
- Regularly review and update the mapping matrix to ensure that it remains accurate and complete.



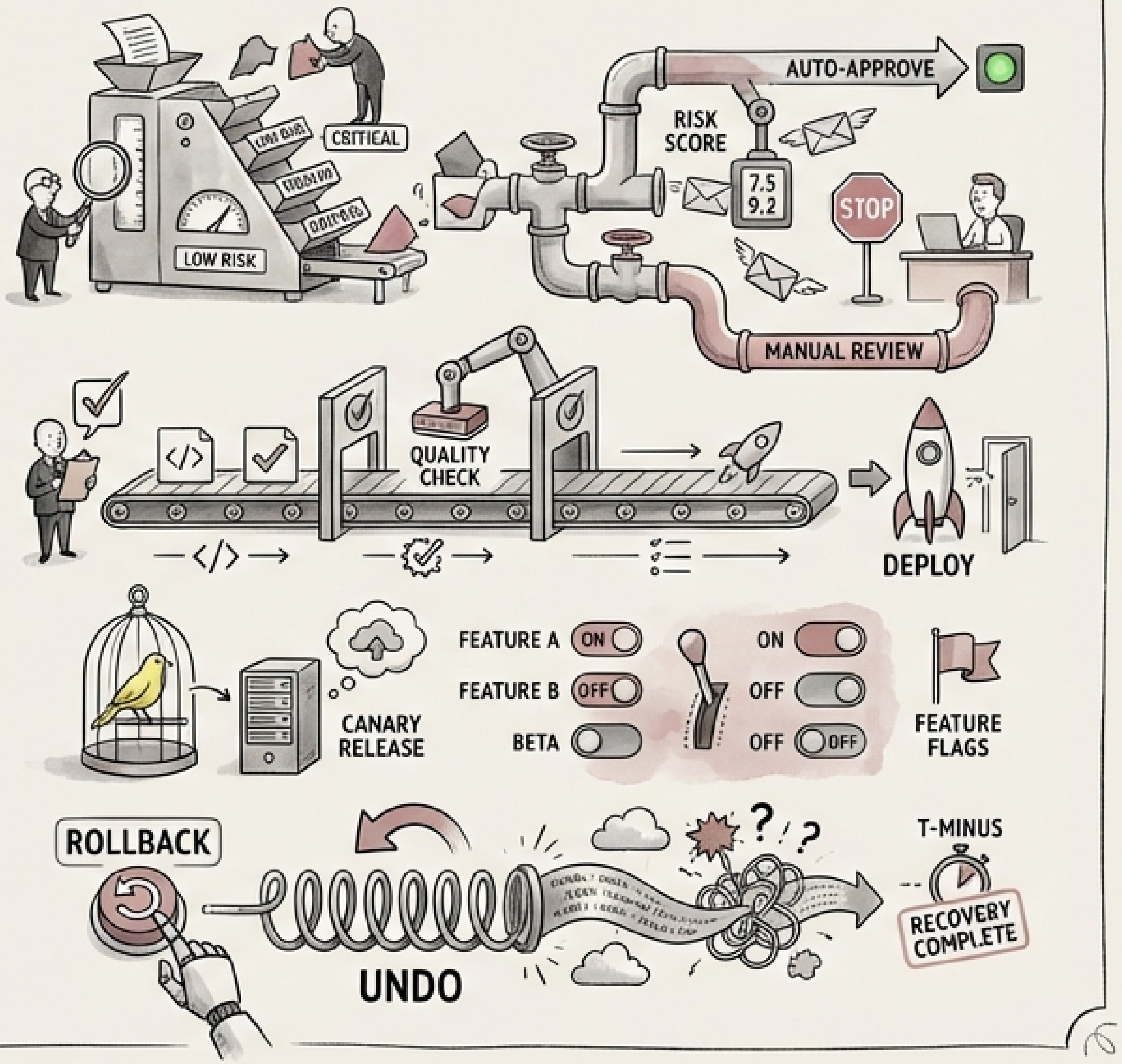
Key Takeaways: Embracing Automated Governance for AI-Driven Development



- **Automated governance** enables **faster deployment** cycles while maintaining **quality** and **security.**
- **Risk-based routing** and **AI-assisted risk assessment** streamline the **change approval** process.
- Comprehensive release gates and deployment strategies **minimize the risk** associated with new releases.
- **Automated rollback and recovery** procedures ensure rapid response to incidents.
- **Compliance evidence** from **CI/CD** automates audit preparation and reduces compliance costs.

Next Steps: Implementing Automated Change Management in Your Organization

- Start by defining clear change classifications and risk assessment criteria tailored to your environment.
- Implement automated risk scoring and routing to streamline the change approval process.
- Integrate release gates into your CI/CD pipeline to ensure quality at every stage.
- Adopt deployment strategies like canary releases and feature flags to minimize risk.
- Automate rollback and recovery procedures to ensure rapid incident response.



Thank You

- Questions?