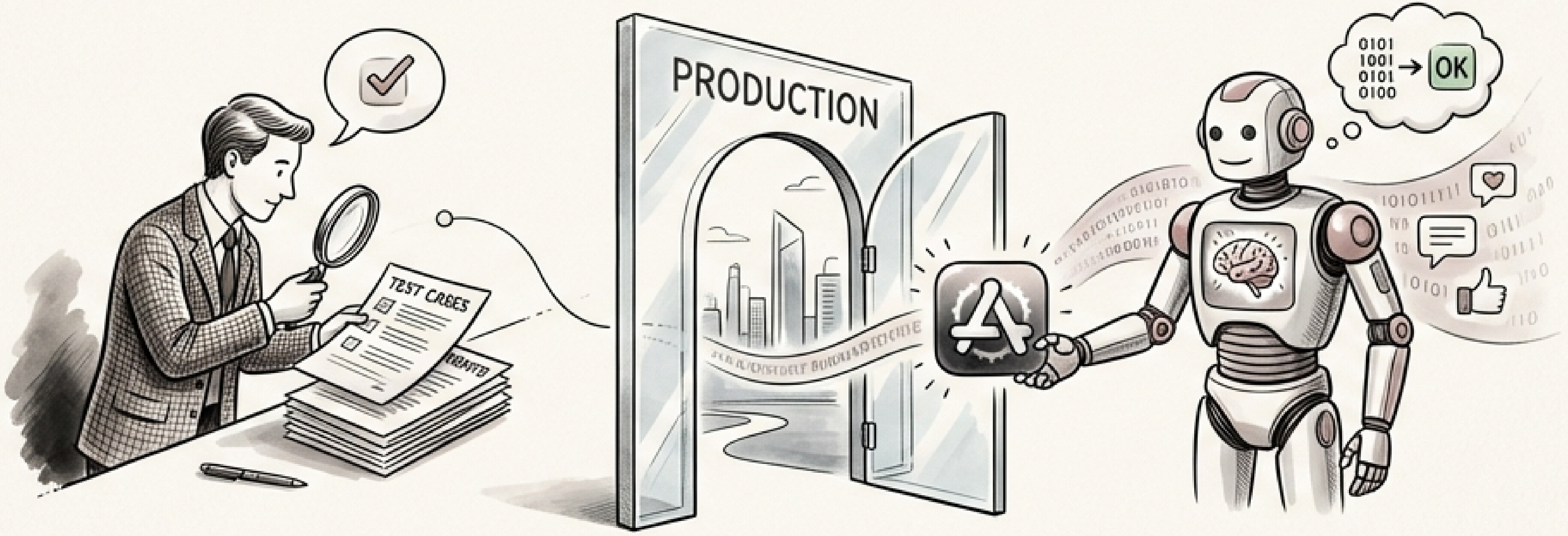
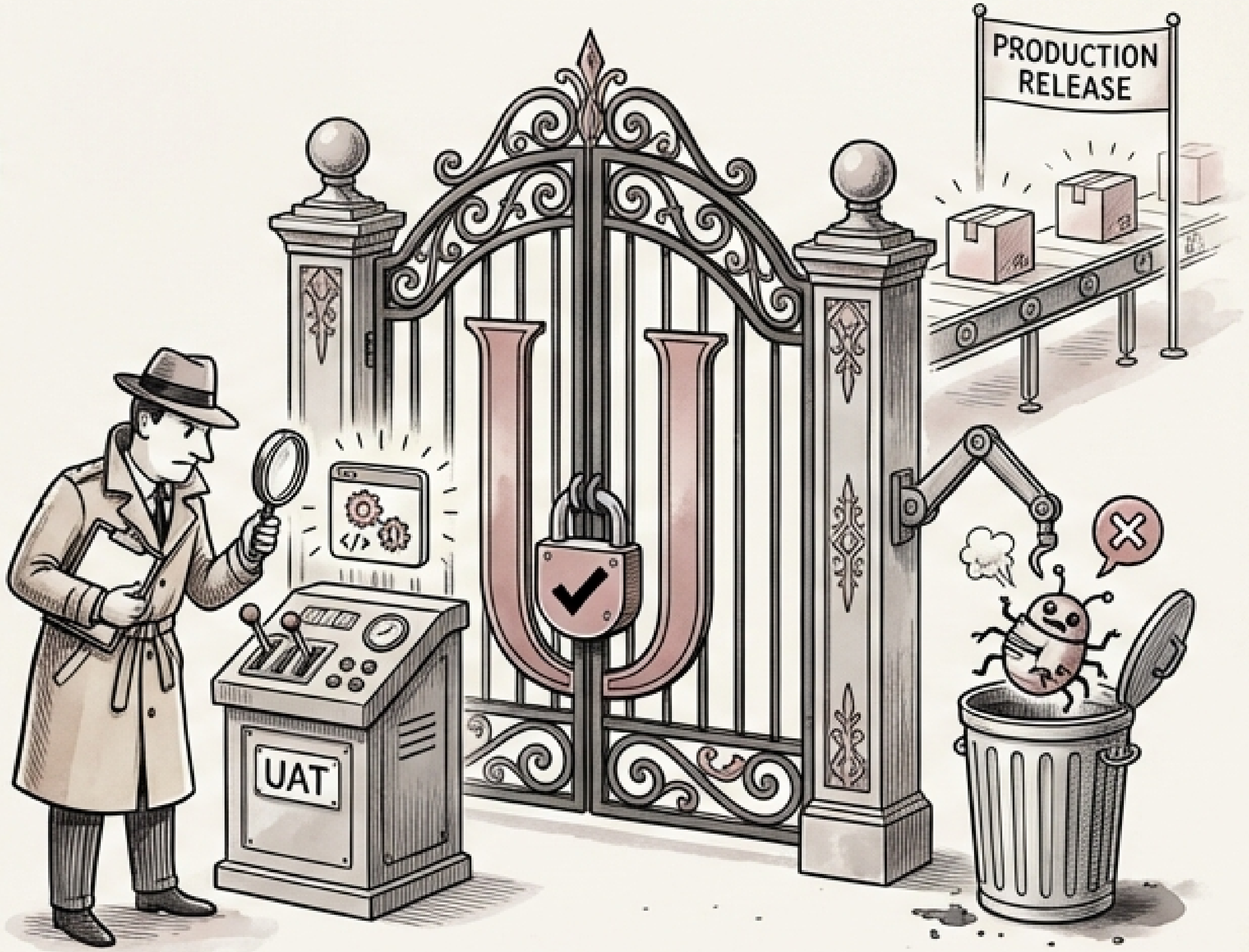
UAT & Acceptance Testing: Validating Software Before Production in AI-Augmented Development

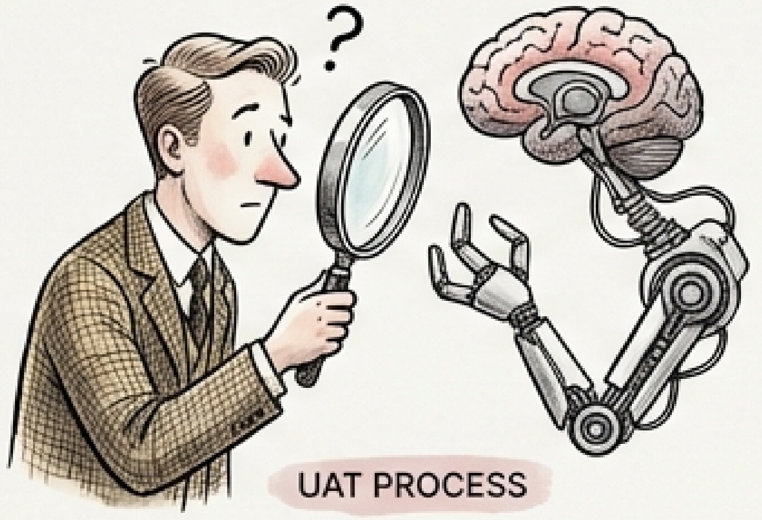


UAT: The Final Gatekeeper Before Production Release

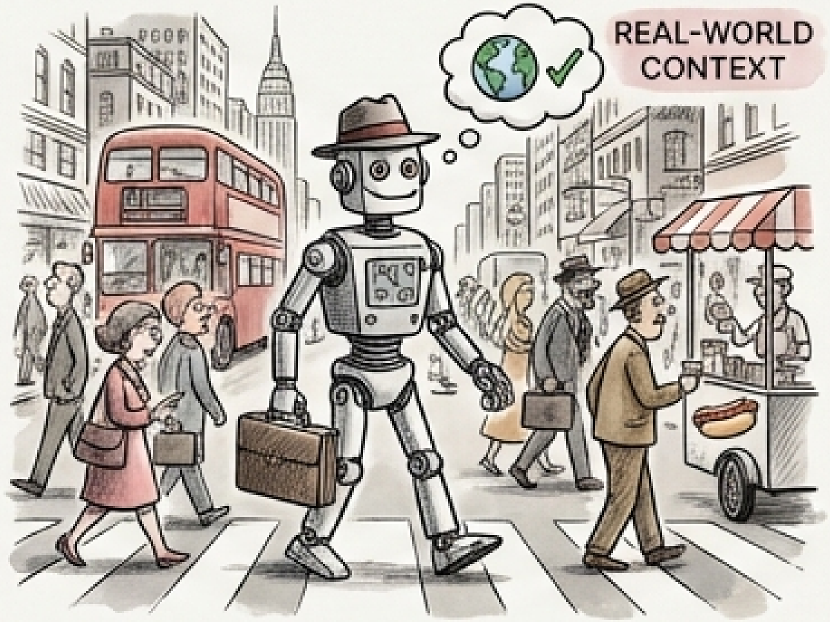
- User Acceptance Testing (UAT) is the ultimate validation step.
- Confirms the system meets stated business requirements.
- Ensures software is truly fit for its intended purpose.



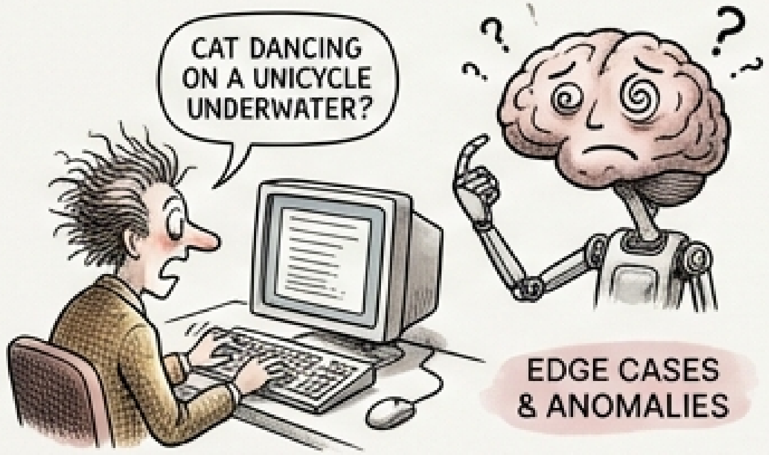
Critical: Validating AI-Generated Functionality in UAT



- UAT must specifically validate AI components.

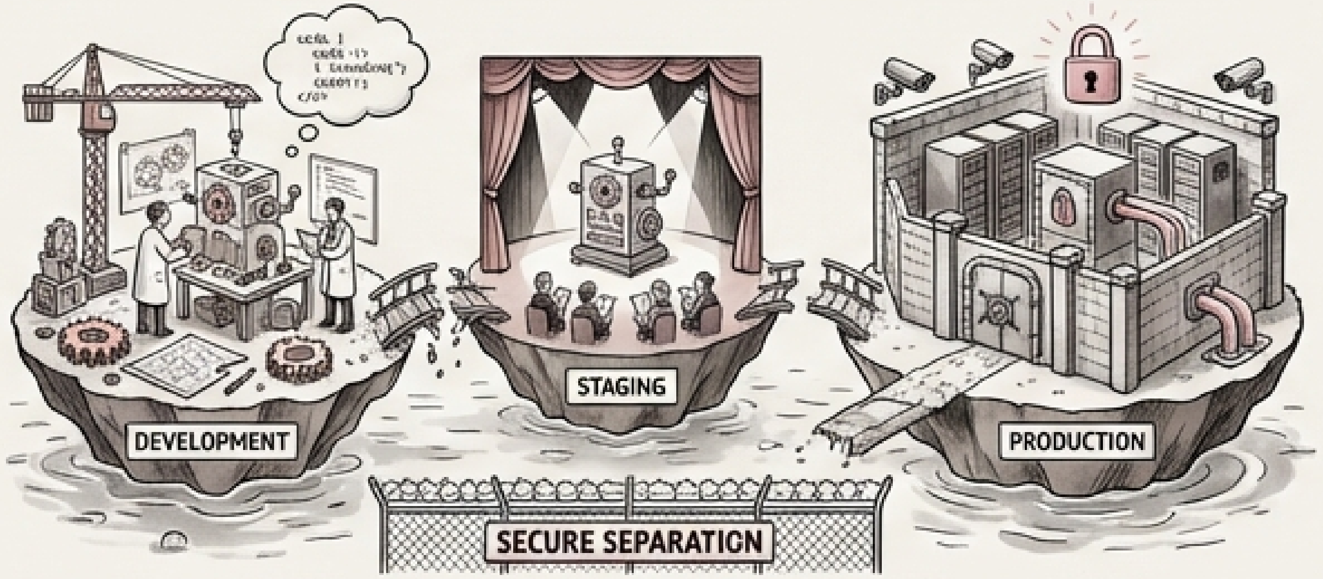


- Ensures AI behaves correctly in **real-world** scenarios.

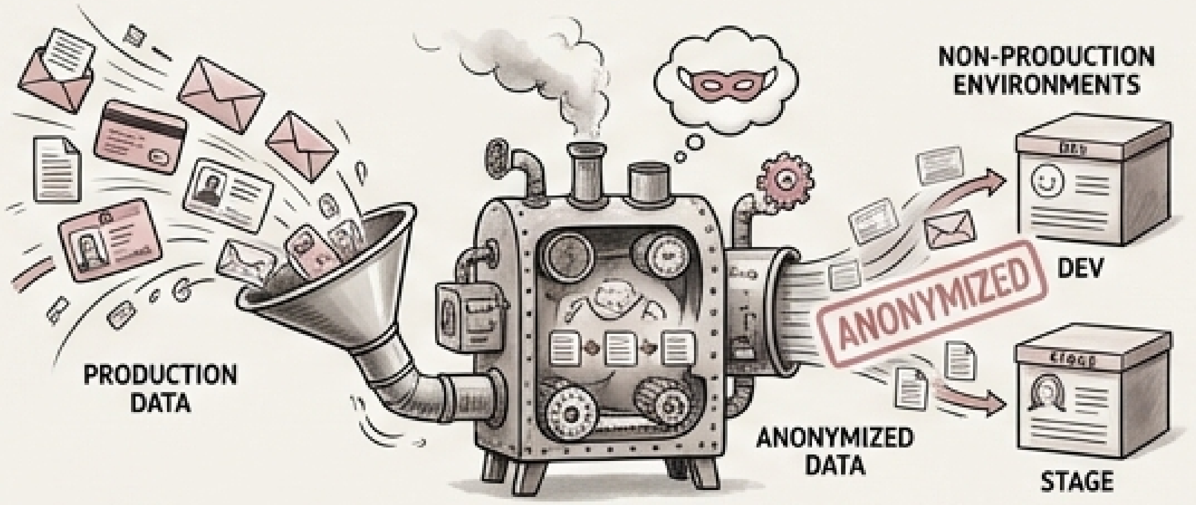


- Focus on unpredictable **user behavior and** edge cases.

Environment Separation: A Non-Negotiable Security Imperative



- Strictly separate development, staging, and production environments.



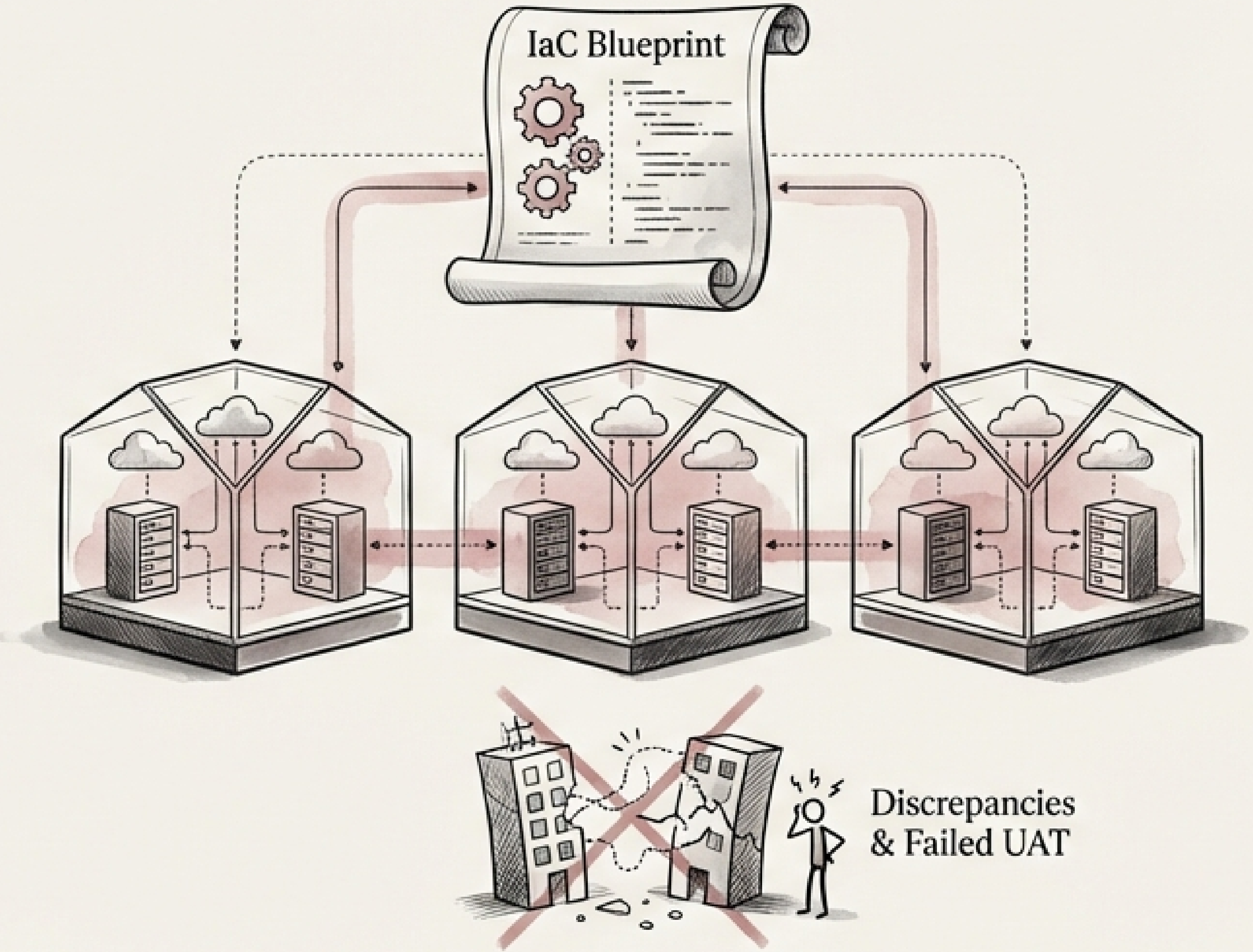
- Prevents accidental data breaches and compliance violations.

Guaranteeing Environment Parity with Infrastructure-as-Code

- Infrastructure-as-Code (IaC) ensures consistent environments

- Same configuration, dependencies, and network topology across environments

- Reduces discrepancies that lead to failed UAT



The New Yorker Style • Presentation Slide

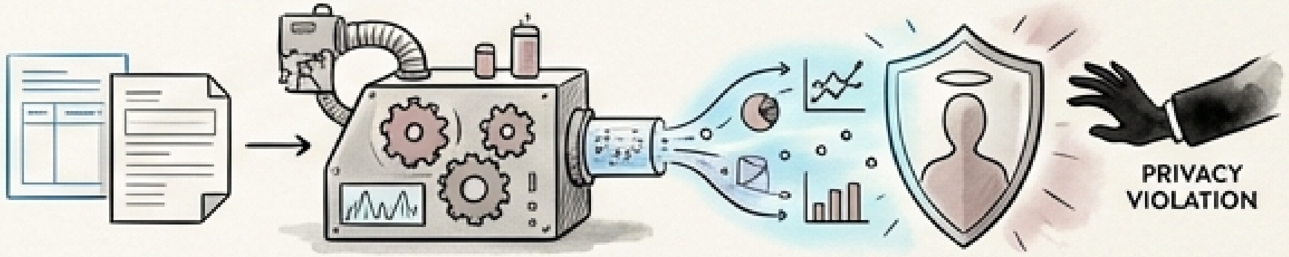
Secure Testing with Data Anonymization and Synthetic Data

DATA NALYST
TESTING ENVIRONMENT
PSEUDONYMIZATION
DATA MASKING
GENERALIZATION
K.ANONYMITY
SYNTHETIC DATA

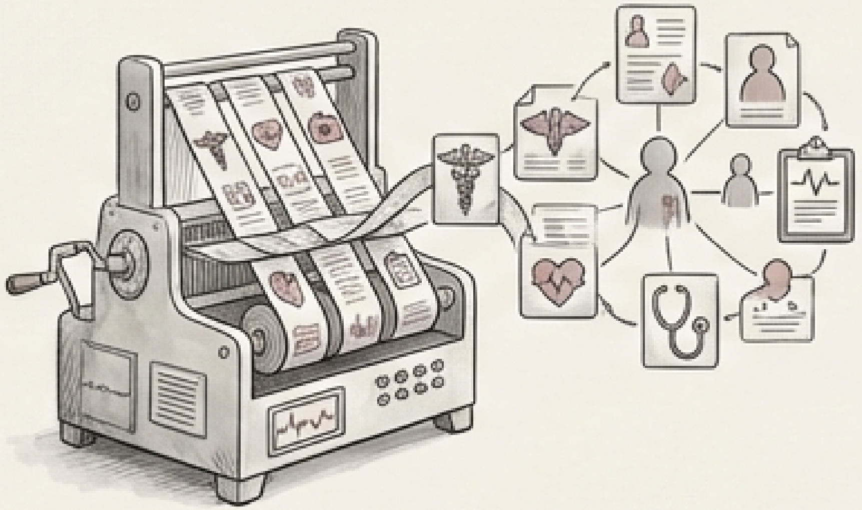
- Production data in testing environments creates compliance risks.

- Use anonymization techniques: pseudonymization, data masking, generalization, k-anonymity.

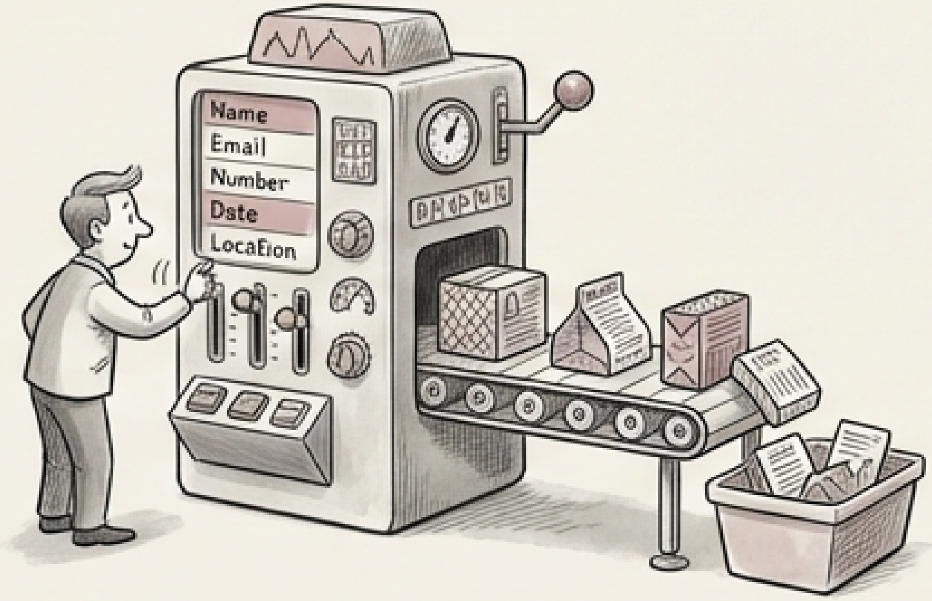
- Generate synthetic data to avoid privacy violations.

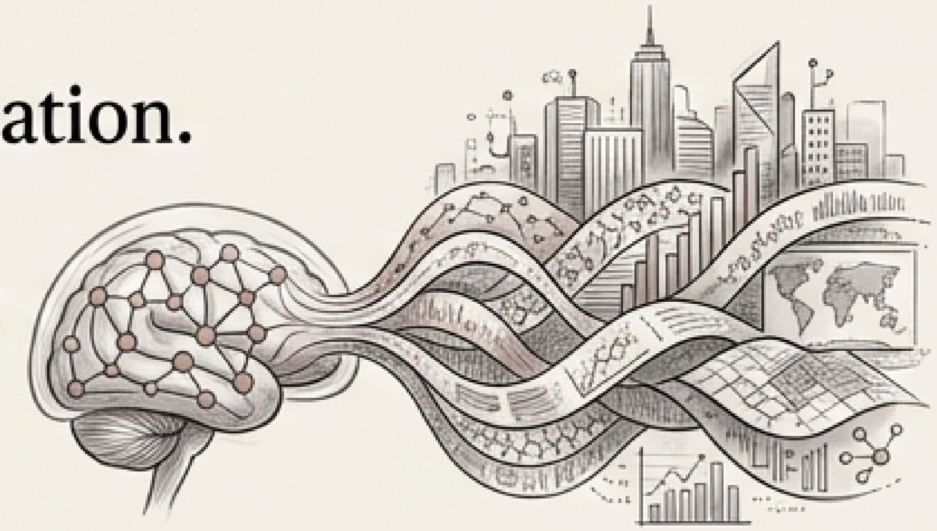


Tools for Synthetic Data Generation



- Faker libraries provide realistic sample data.
- Synthea creates synthetic healthcare data.

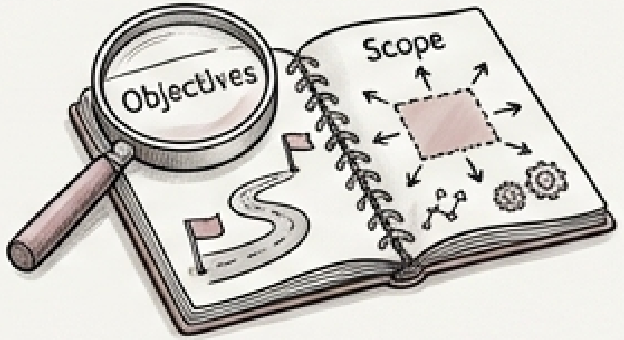


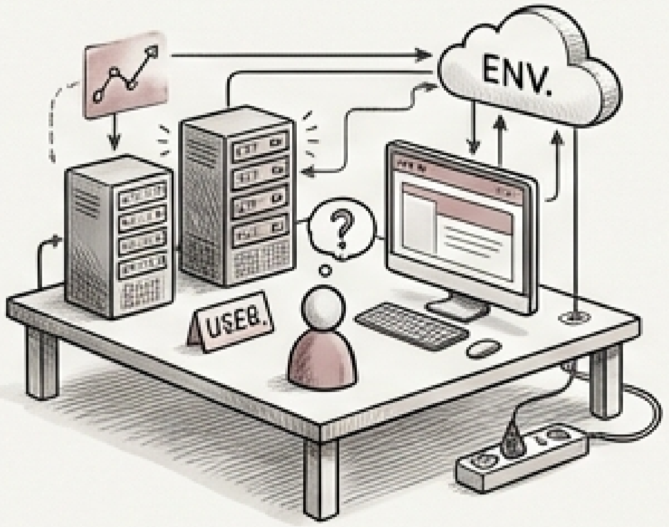


- Mockaroo offers customizable data generation.
- AI can generate complex synthetic datasets.

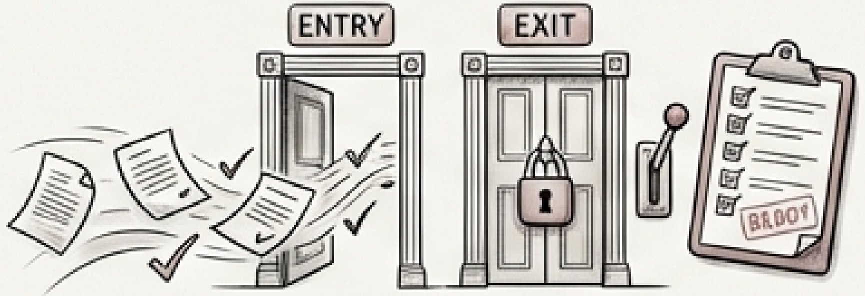
The 9 Essential Sections of a Complete UAT Test Plan

- 1. Scope and Objectives

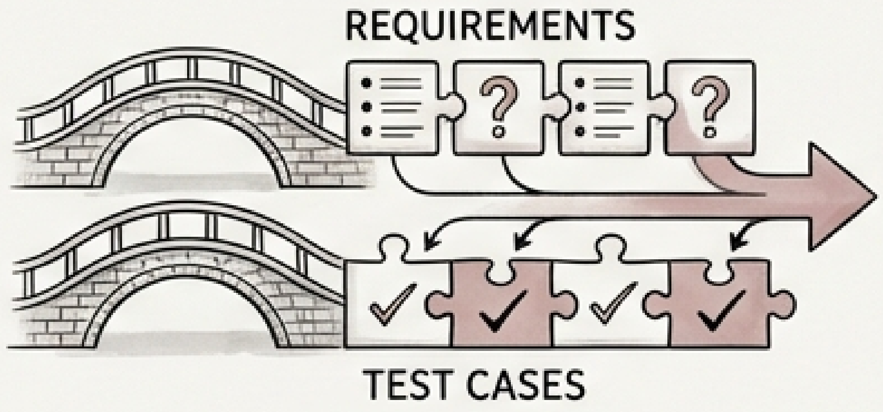


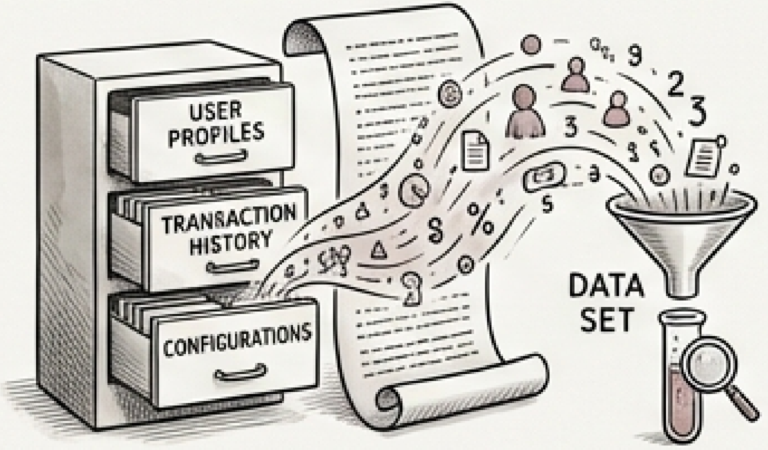


- 2. Test Environment Requirements



- 3. Entry and Exit Criteria
- 4. Test Cases Mapped to Requirements

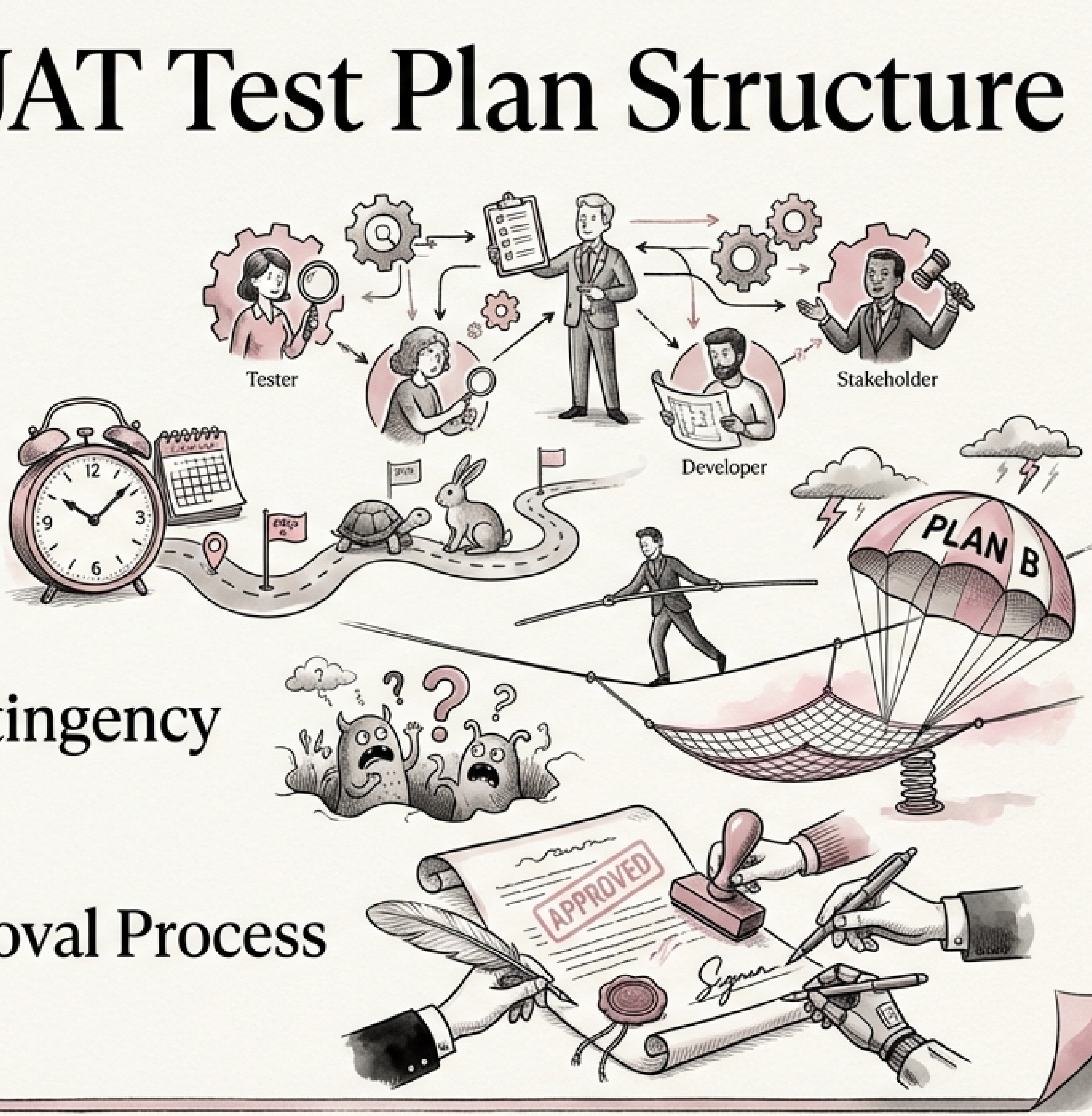




- 5. Data Requirements

Completing the UAT Test Plan Structure

- 6. Roles and Responsibilities
- 7. Schedule and Milestones
- 8. Risk Assessment and Contingency
- 9. Sign-off Criteria and Approval Process

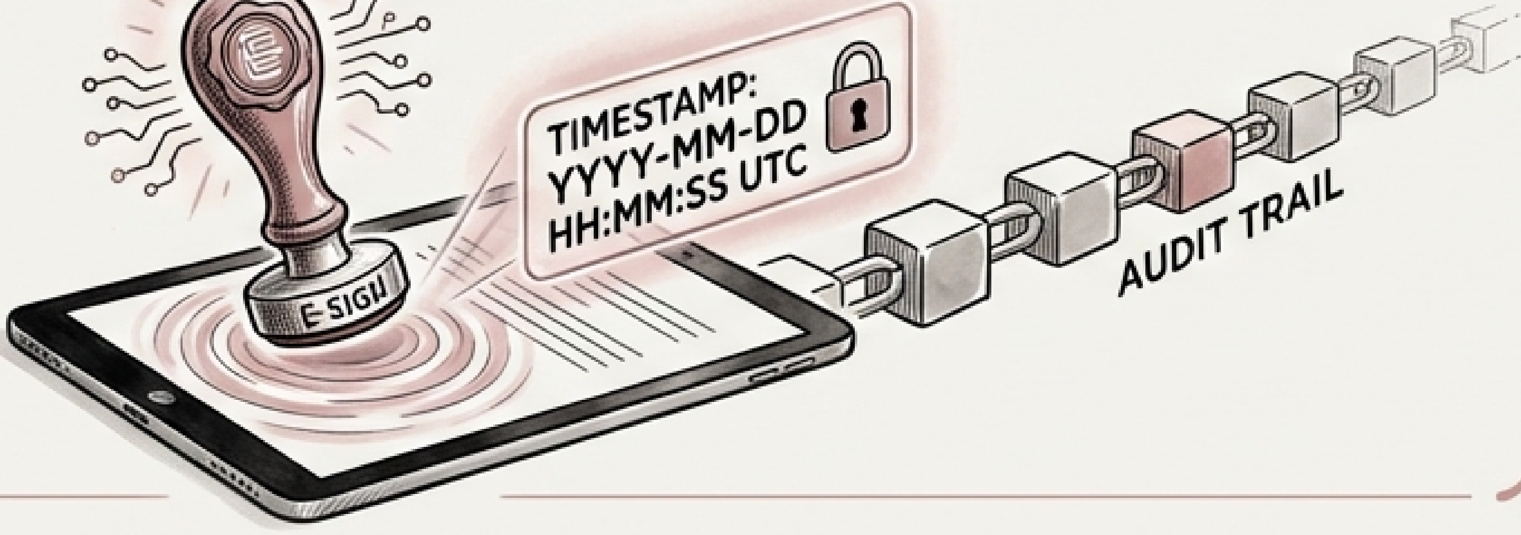


FORMAL SIGN-OFF: MAINTAINING AN AUDIT TRAIL FOR COMPLIANCE

- **Designate sign-off authority:** business owner, security representative, compliance officer.

- **Implement digital sign-off** with timestamps for an immutable audit trail.

- **Compile evidence package:** test results, defect status, **coverage metrics, security scans.**



UNDERSTANDING CONDITIONAL SIGN-OFFS

- 1. Document accepted risks when using conditional sign-offs.

- 2. Define clear remediation timelines for identified issues.

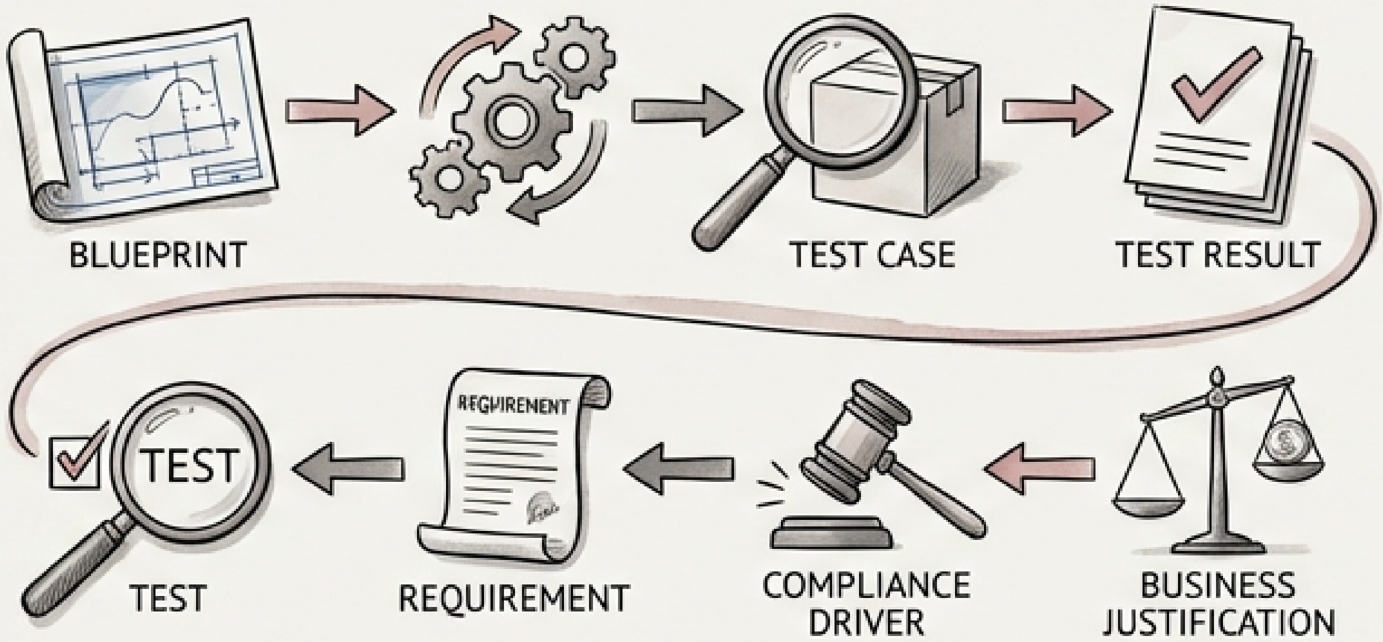
- 3. Establish ongoing monitoring requirements to track risk mitigation.

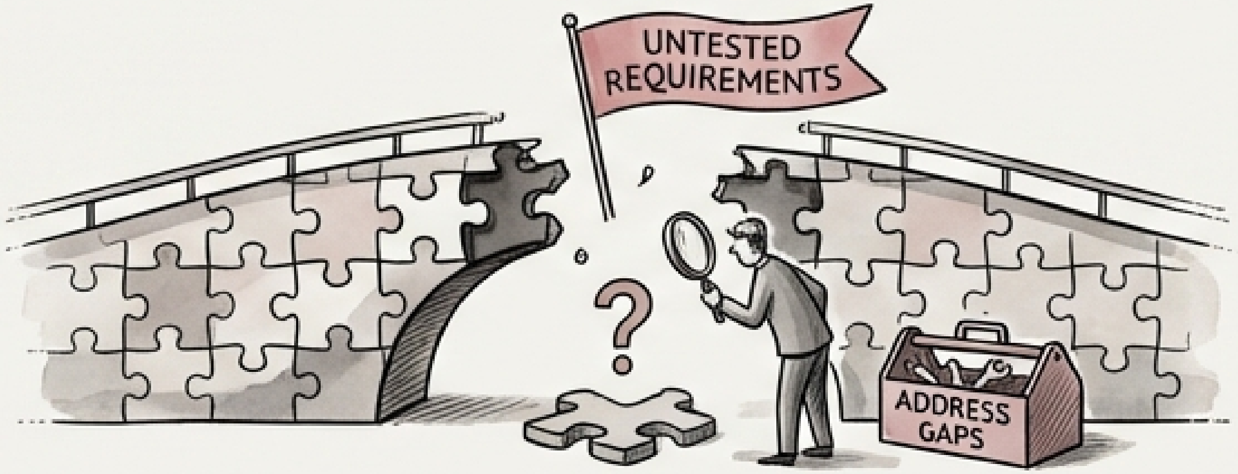
Bidirectional Traceability: Ensuring Comprehensive Test Coverage

- Trace every requirement forward: design → implementation → test case → test result.

- Trace every test backward: requirement → compliance driver → business justification.

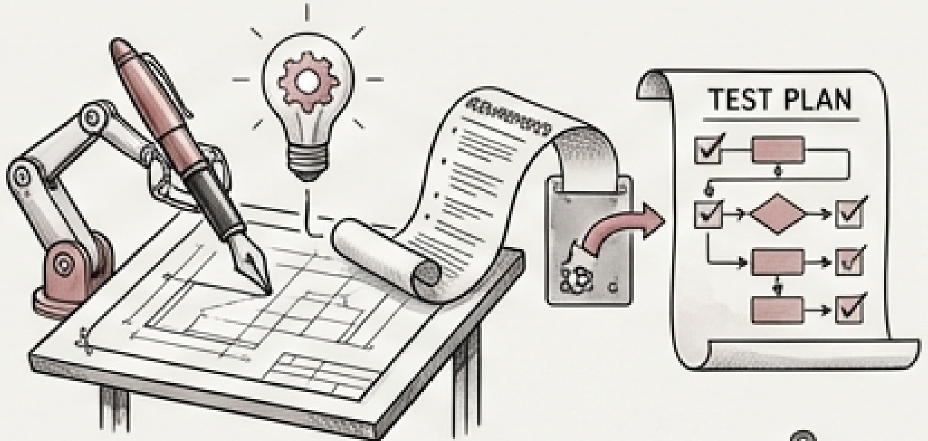
- Identify and address gaps in traceability indicating untested requirements.

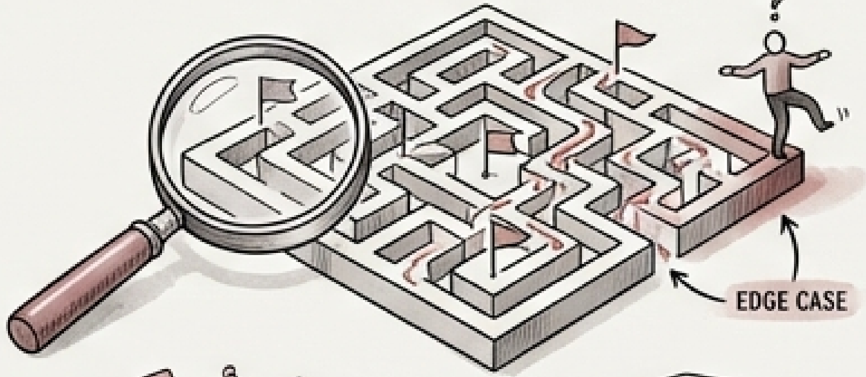


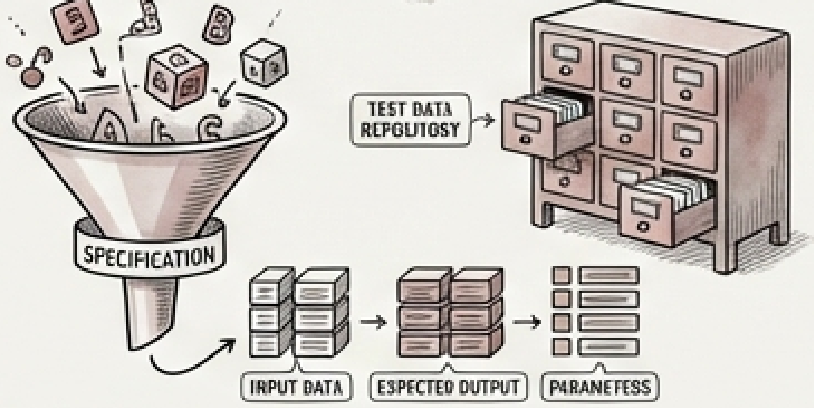


AI-Assisted Test Plan Generation: Boosting Efficiency

- AI tools can generate initial test plans from requirements.
- Useful for comprehensive test case generation and edge case identification.
- Aids in specifying data requirements for testing.





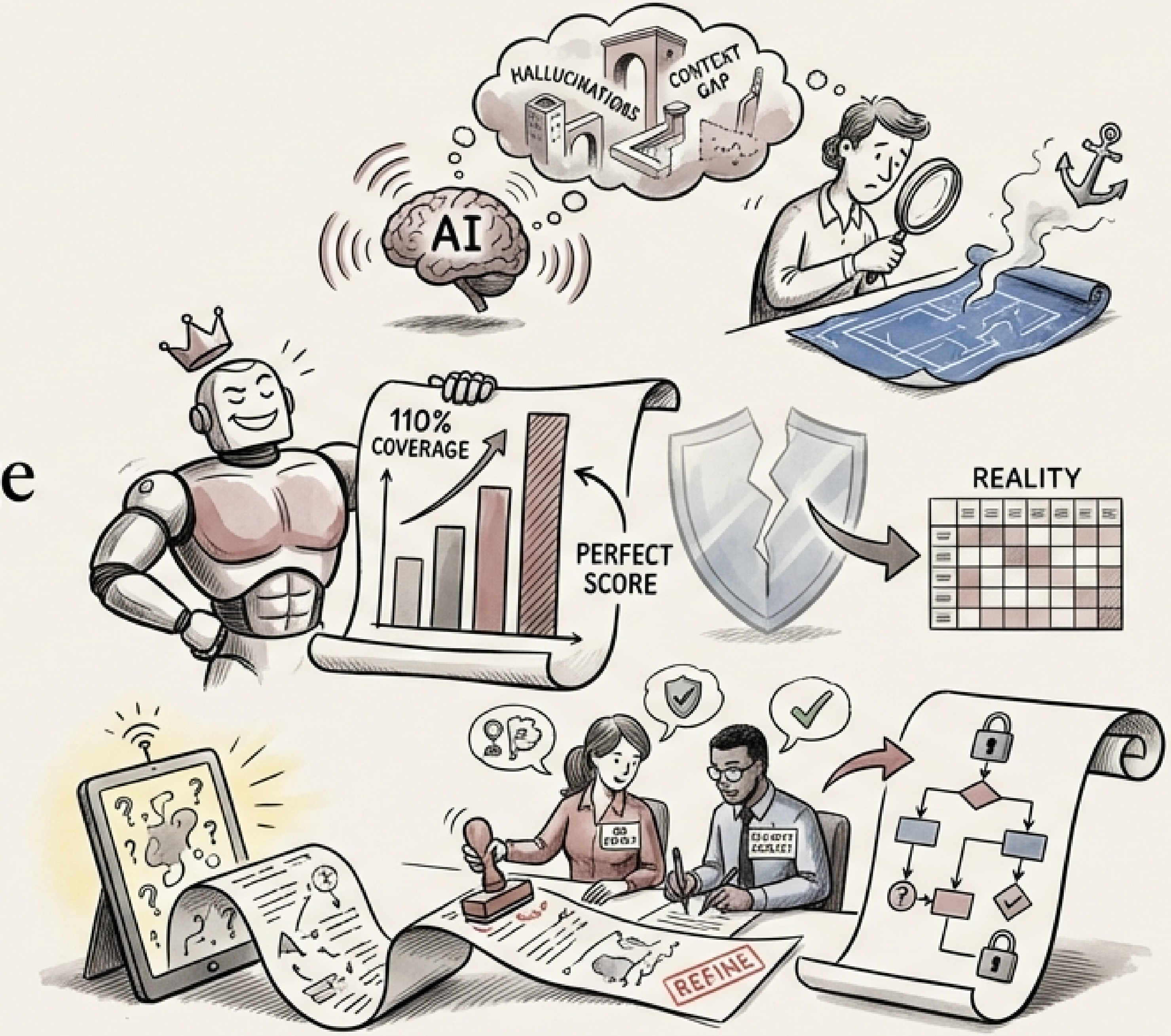


Mitigating Risks of AI-Generated Test Plans

- Guard against hallucinated requirements and missing context-specific scenarios.

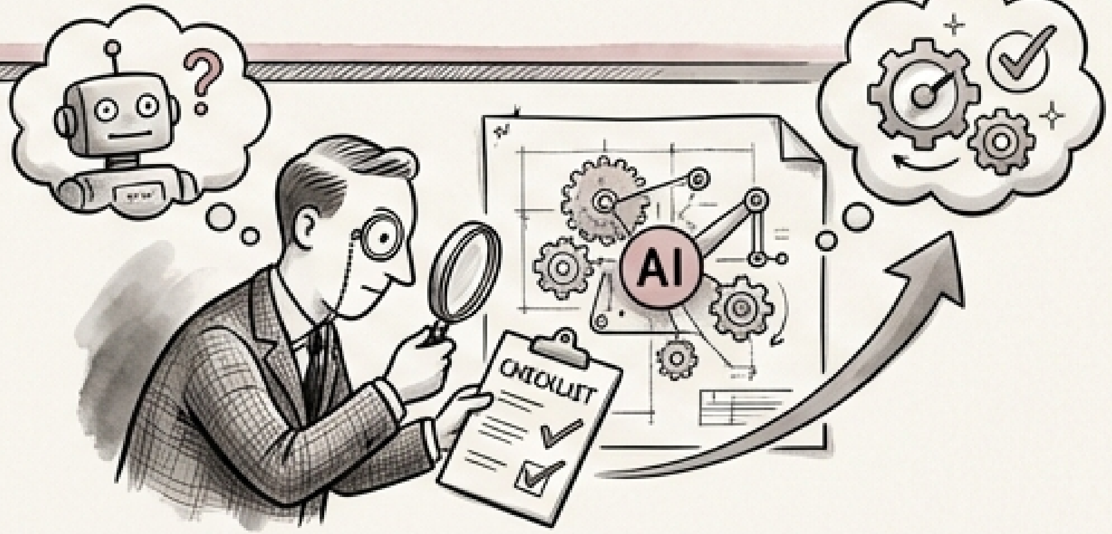
- Avoid overconfident coverage claims from AI.

- QA and security teams must review and refine AI-generated drafts.



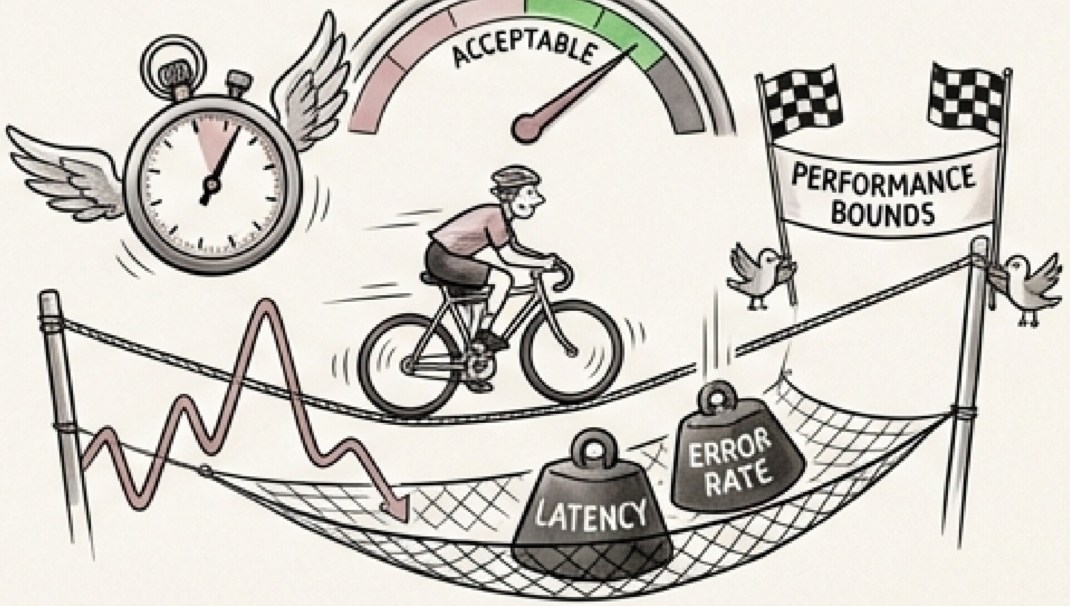
DEFINING ACCEPTANCE CRITERIA FOR AI-GENERATED FEATURES

- Verify functional correctness independently of the AI model.



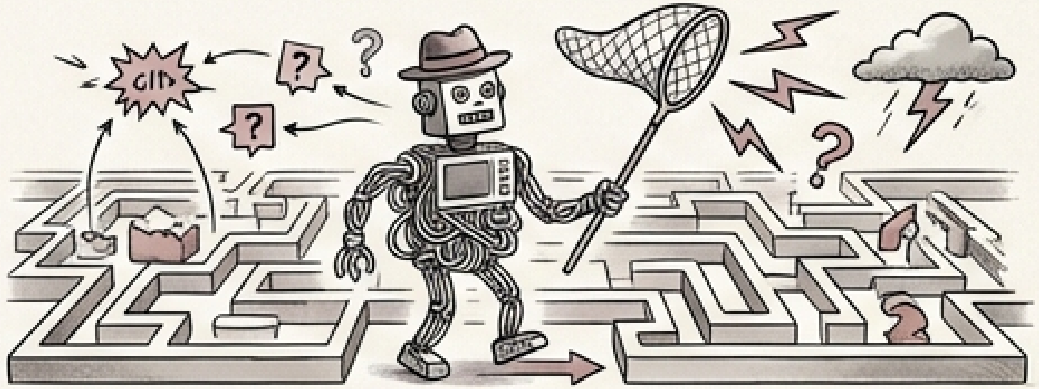
- Ensure security requirements are met beyond just functional correctness.

- Confirm performance is within acceptable bounds.



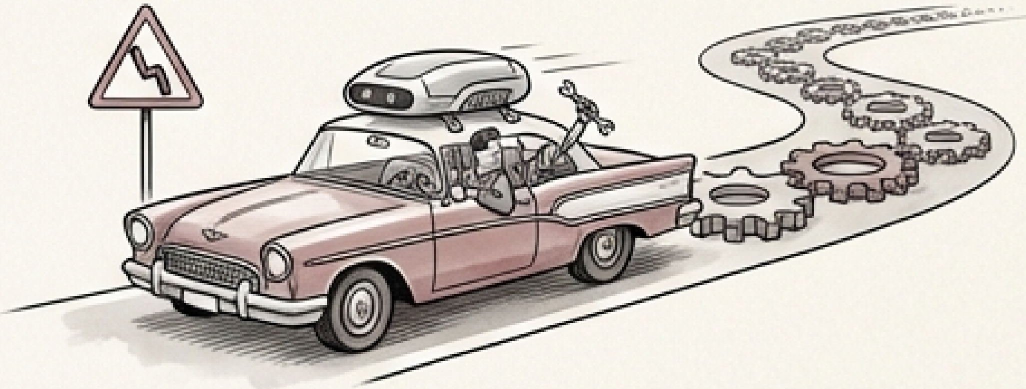
AI Feature Acceptance: Error Handling and Accessibility

- Verify robust error handling for all edge cases and unexpected inputs.



- Maintain accessibility compliance (e.g., WCAG) for all users.

- Prevent regressions in existing functionality when introducing AI features.



Thank You

- Questions?