



**UTT**

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

**GOBIERNO DE BAJA CALIFORNIA**

**TOPIC:**

Techniques and tools for tracking test.

**PRESENTED BY:**

Padilla Virgen Jorge Luis

**GROUP:**

10B

**SUBJECT:**

Software Development Process Management

**TEACHER:**

Ray Brunett Parra Galaviz

Tijuana, Baja California, January 13th 2025

## **Techniques and Tools for Tracking Evidence in Testing**

### **Definition**

Tracking evidence in software testing involves monitoring, documenting, and managing the progress and results of tests to ensure quality assurance and accountability. It provides transparency, identifies potential issues early, and ensures compliance with project requirements.

### **Techniques for Tracking Evidence**

#### **1. Test Case Management:**

- Organize test cases into categories (functional, performance, security) and track their status (e.g., passed, failed, in progress).
- Use traceability matrices to link test cases with requirements or defects.

#### **2. Defect Tracking:**

- Document and prioritize bugs discovered during testing.
- Track their lifecycle: reported, assigned, resolved, and verified.

#### **3. Automation Logs:**

- Utilize logs generated by automated test tools to track executed tests and their outcomes.

#### **4. Test Coverage Analysis:**

- Measure the extent of code or functionality covered by tests to ensure comprehensive testing.

#### **5. Version Control Integration:**

- Link test artifacts to specific versions of the codebase to track changes and regression issues.

## 6. Metrics and KPIs:

- Track metrics like test pass rate, defect density, mean time to detect (MTTD), and mean time to resolve (MTTR).

## Tools for Tracking Evidence

### 1. Test Management Tools:

- **Jira**: Provides test management plugins for organizing and tracking tests.
- **TestRail**: Offers a centralized platform for managing test cases, plans, and results.
- **Zephyr**: Integrates with Jira for real-time test tracking.

### 2. Defect Tracking Tools:

- **Bugzilla**: A robust tool for logging and tracking bugs.
- **MantisBT**: A simple platform for managing issues.
- **Redmine**: Offers defect tracking and project management capabilities.

### 3. Automation Testing Tools:

- **Selenium**: Provides detailed logs for automated tests.
- **Cypress**: Tracks automated test execution and generates reports.

### 4. Continuous Integration (CI) Tools:

- **Jenkins**: Integrates with test frameworks to track automated test results in CI pipelines.
- **GitHub Actions**: Logs test results tied to specific commits.

### 5. Code Coverage Tools:

- **SonarQube**: Analyzes code quality and test coverage.
- **JaCoCo**: Tracks Java test coverage.

## 6. Traceability Tools:

- **ALM/Quality Center:** Tracks requirements, test cases, and defects in a single platform.
- **ReqTest:** Links test cases and results to project requirements.

## Best Practices for Tracking Evidence

### 1. Centralized Repository:

- Store all test artifacts, results, and logs in a single location for easy access and traceability.

### 2. Real-Time Updates:

- Use tools with real-time synchronization to ensure up-to-date information.

### 3. Clear Documentation:

- Maintain detailed records of test cases, defects, and resolutions to provide an audit trail.

### 4. Regular Reporting:

- Generate periodic test reports to share progress with stakeholders.

### 5. Collaboration:

- Ensure testers, developers, and managers have access to the tracking system to improve communication and accountability.

## Benefits

- **Transparency:** Provides clear visibility into testing progress and outcomes.
- **Accountability:** Tracks who performed which tests and when.
- **Improved Quality:** Identifies gaps in coverage and areas requiring more focus.
- **Risk Management:** Ensures timely detection and resolution of critical defects.
- **Compliance:** Meets regulatory requirements by maintaining detailed test records.