

MAJOR HEALTH INSURANCE CARRIER FULLY AUTOMATES THE DELIVERY OF SYNTHETIC TEST DATA

The Client

A major health insurance company operating in more than a dozen US states offering both government and privately funded healthcare plans and serving tens of millions of subscribers.

Business Challenge

The insurance company needed to ensure accuracy in its claims processing systems.

- The accuracy of insurance claims processing is an industry-wide problem with estimates of up to 80% of submitted claims having errors resulting from human error or deliberate fraud.
- **Inaccurate claims result in overcharges and revenue leakage for healthcare payers and delayed payments for providers.** The insurance carrier in this case study decided to invest in new systems to dramatically reduce errors and streamline the entire claims processing cycle.

As part of their digital transformation, the company is migrating from a monolithic in-house claims processing system to a modern distributed system to improve speed, accuracy, efficiency, and security.

- The company decided to undertake a major in-house software development effort to achieve all these goals with cutting edge development tools and a fully automated software test environment.
- **It was mandated that no Protected Health Information (PHI) would be used in any phase of the software development and testing process.** As a result, a comprehensive synthetic test data solution became a critical component of this software migration initiative.

Requirements

- To accurately simulate the many types of data exchanged during a claims processing workflow, the GenRocket platform was deployed to generate synthetic transaction data in the **X12 EDI** format to deliver 100% HIPAA compliant test data.
- The GenRocket platform supports both the current X12 EDI 837 standard (5010) and the next generation (8020) that covers expanded transaction sets for wearable medical devices and IoMT technologies.
- A typical claims data feed contains a complex assortment of data for many subscribers having many types of treatments rendered by multiple providers.
- Such feeds needed to be tested under various conditions to ensure they function properly.

The Solution

GenRocket Synthetic Test Data

X12 EDI XSD

- Readily available X12 EDI XSD templates to generate the volume or variety of synthetic data in the proper format for any category of software testing.

CI/CD Pipeline Integration

- Automate the delivery of synthetic data directly into a CI/CD pipeline to accelerate every stage of the software testing lifecycle.

Center of Excellence (CoE) Model

- Created a small team of test data engineers fully trained on the four-stage GenRocket methodology: Model, Design, Deploy, and Manage.
- CoE is responsible for modeling and designing all *Test Data Cases* (the instruction sets used by GenRocket to control the synthetic data generation process at runtime).
- Once designed, *Test Data Cases* are called and executed by test case scripts developed for each stage of automated testing.

9-Step Implementation Framework

To ensure thorough and accurate testing of its health claims processing network, the insurance company focused on the following testing best practices utilizing GenRocket synthetic test data.

1. Test Early (Shift Left)

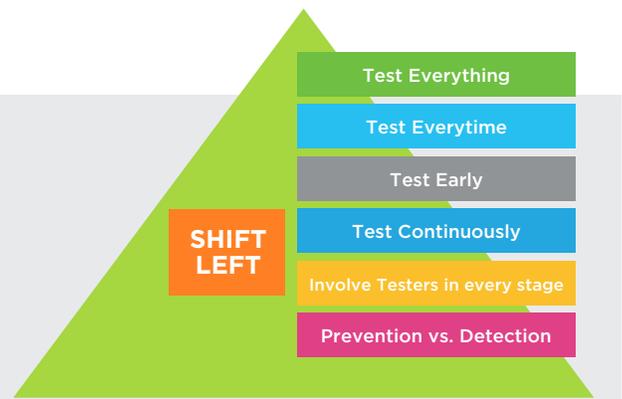
Testing early places a strong emphasis on unit testing to catch defects early when they are easier and less costly to resolve. Synthetic datasets can be designed in small increments so they can be aggregated into larger and more sophisticated Test Data Cases as the testing lifecycle progresses. It also allows functional testing to be performed at the module level and allows system testing to focus on integration issues.

TDM Tool and Framework - Quality Benefits

Testing Principle - 1: Test Early (Shift Left)

Benefits: Early detection, saves cost

- Identify test data requirements as early as possible
- Identify test data as early as possible
- Generate claims and share with developers
- Do as much testing as possible during Unit Test
- QA should support Unit Test (Helps QA learn from Developers)
- Helps understand the scope of the deliverables among business, development and QA



2. Thorough Regression Testing

Regression testing is critical to prevent defect leakages. This saves time and money. The insurer reused GenRocket's Test Data Cases to allow for full regression testing as each new feature is developed and added to the release pipeline.

Regression Testing



TDM Tool and Framework - Quality Benefits

Testing Principle - 2: Thorough Regression

Benefits: Prevents defects leakage, Quality Deliverables, Cost Saving

- Start building test data cases with the first story
- Build Test Data Cases incrementally as feature development progresses
- Aggregate Test Data Cases into Chapters, Stories, and Epics
- Generate claims and test for each increment
- Conduct full regression testing at each stage

3. Test Environment Configuration

Test environment configuration played an important part in this solution. Three separate test environments were deployed for testing code as it progresses up the software testing pyramid. There is also a final staging area to provide a production look-alike environment to simulate real-world behavior. This enabled the insurer to identify issues that may be related to the configuration of each test environment.



TDM Tool and Framework - Quality Benefits

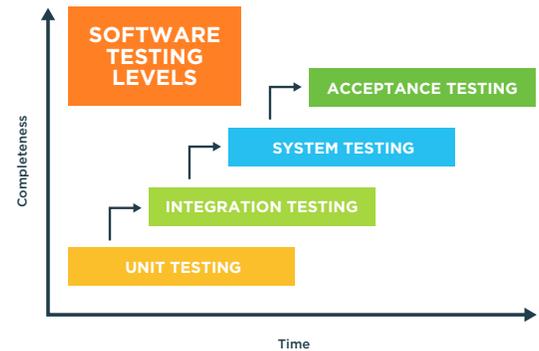
Testing Principle - 3: Identify Issues Related to Environment Configuration

Benefits: Go or No-Go Decision, scope definition

- Execute Test Data Cases as code is promoted to upper-level environments
- Identify and resolve any configuration issues
- Validate test environment readiness

4. Performance Test at Every Level

Because GenRocket's Test Data Cases are reusable, the insurer can performance test at every level. Any synthetic data design can be quickly modified to add more records using the same data structure and data generation rules. This allows performance issues to be detected early and at every stage of development. The practice of repeating tests with higher data volume increases confidence in the quality of the code and predictability of how it will behave at various load and usage scenarios.



TDM Tool and Framework - Quality Benefits

Testing Principle - 4: Performance Test at every test level

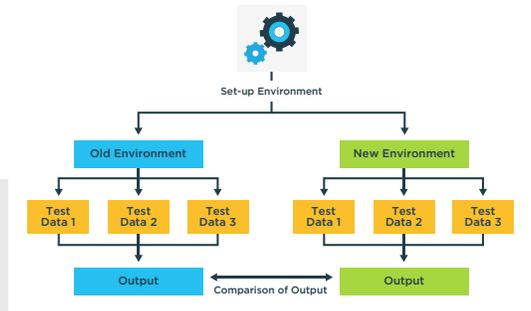
Benefits: Early Detection and Prevention, Time, Resource and Cost Saving

- Reuse previously executed Test Data Cases as much as possible
- Add additional members and providers to achieve volume of claims
- This may be seen as redundant testing.
- It's an overlapping testing to prevent surprises.
- Increases tester confidence
- Increases predictability



5. Parallel Test at Every Level

Because this insurance company is migrating from a legacy application to a new claims processing system, it's important to test both systems in parallel at every stage of development. This identifies any functional discrepancies or missing requirements during the early stages.



TDM Tool and Framework - Quality Benefits

Testing Principle - 5: Parallel Test at every test level

Benefits: Identify functionality discrepancies in early stage, identify missing requirements, identify enhanced functionality in new system.

- Reuse existing Test Data Cases
- Send claims to both systems simultaneously
- Identify discrepancies and review with development team
- Document resolution decisions for future reference.

6. Reuse Test Data Cases

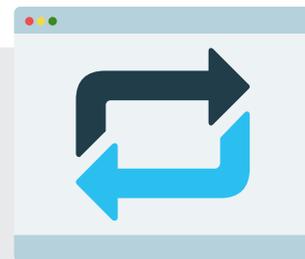
Reusability is fundamental to the achieving cycle time acceleration. It allows for the centralized design and management of Test Data Cases so they can be deployed using a distributed self-service solution for DevOps teams. The insurance company uses a TDA framework where all test data requests submitted by users, along with the *Test Data Cases* created by CoE staff, are stored in a library that is both categorized, and revision controlled.

TDM Tool and Framework - Quality Benefits

Testing Principle - 6: Reusability

Benefits: Increases Efficiency and Productivity

- Reuse existing Test Data Cases
- Categorize in a centrally accessible library
- Version control all Test Data Cases
- Enable a searchable library for distributed access



7. Traceability of Test Results

All tests conducted by the TDA framework are tracked by measurable metrics and are fully traceable. This allows for a quality audit of the entire application development process. All Test Data Cases are linked to every test case script as well as the results of each test execution. Traceability ensures the testing process is always fully validated and documented.

TDM Tool and Framework - Quality Benefits

Testing Principle - 7: Traceability

Benefits: Measurable Metrics, Quality Audit

- Link Test Data Cases with test cases and test executions
- The linkage can itself be proof for test evidence.



8. Test with Valid Data

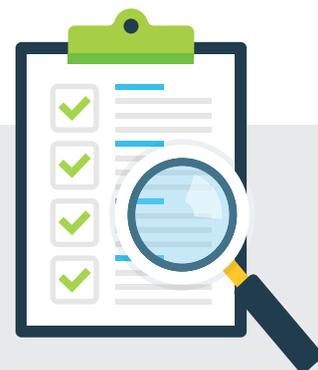
At the final stage of testing, a user interface testing tool is used to test the fully integrated system to validate outcomes and compare them with expected results. Testing with valid data at this stage ensures predictability of the software under the widest possible variety of conditions and simulates real-world operation of the code.

TDM Tool and Framework - Quality Benefits

Testing Principle - 8: Test with valid data

Benefits: Predictable outcome

- Many variables can impact the claim processing and expected outcome.
- Testing with a compatible claim data is more important to achieve predictable outcome (expected result)
- UI testing tool will help identifying valid/compatible claim data.



9. Measure Test Results

Collecting test evidence makes all data available for audit purposes and captures important quality metrics for the entire software testing process.

TDM Tool and Framework - Quality Benefits

Testing Principle - 9: Test Evidence

Benefits: Quality Metrics, Test Evidence for Audit Purpose

- Test Data Cases with test data can be used for test evidence.



Outcome

The Test Data Automation Framework established by this insurance carrier provides a scalable deployment model for distributed self-service as it incorporates Agile testing best practices across the entire development lifecycle. As a result, the DevOps team realized many benefits from this approach including:

- Acceleration of total test cycle time
- Maximized coverage for all software under test
- Full performance and regression testing at every level
- Reusability of test cases and Test Data Cases
- 100% compliance with all HIPAA privacy laws
- Seamless transition to next-generation claims processing systems

The GenRocket solution delivered a clear and sustainable return on investment in the form of dramatic time savings, labor cost reduction, faster time to market, and increased quality of software deployed to production.

