Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Using Requirements Traceability and UAT Controls to Improve Data Quality in Digital Compliance and Billing Systems

Margaret Awuradwoa Nettey
Business Systems Analyst, Bsystems Limited
Greater Accra, Ghana

Professional and Technical Article

ARTICLE INFO

Received: 01 Aug 2021
Revised: 17 Sept 2021
Accepted: 28 Sept 2021

ABSTRACT

This article examines how requirements traceability and user acceptance testing can be applied as practical data-quality controls during the implementation of digital compliance and billing systems. Implementations of these systems frequently produce records that are incomplete, duplicated, or inconsistent with one another, and these defects usually surface late, during reporting, when they are most expensive to correct. The article argues that many such defects can be prevented or detected early by treating requirements traceability as a control that links each business rule to a tested system behaviour, and by designing acceptance testing to exercise data-quality conditions rather than only functional steps. It describes the data-quality issues commonly observed in compliance and billing implementations, sets out how traceability and acceptance testing address them, and proposes a set of control documents and a practical implementation sequence. The intended result is improved data completeness, earlier identification of billing and reporting errors, and compliance and management reports that can be reconciled to their source records.

Keywords: compliance, reconciled, expensive

2. Introduction

Digital compliance and billing systems hold data that other parties rely upon. A regulator reading a compliance return, a finance team recognising revenue, and a manager acting on a report all depend on records they cannot independently verify. The system is treated as the source of truth, so any defect in its records is inherited by every process and decision that draws on them.

In practice, the quality of those records is decided long before go-live, during requirements definition and testing. When the rules that determine whether a record is valid are written down clearly and linked to tests that confirm the system enforces them, defects are caught while they are cheap to fix. When those rules are left implicit, the same defects pass into production and are discovered later, during reconciliation or reporting, when the original context has been lost.

This article sets out two controls that address data quality at this early stage: requirements traceability, which links each business rule to the system behaviour and test that satisfy it, and user acceptance testing designed to exercise data-quality conditions. Both are established parts of business-analysis and

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

software-testing practice, described in bodies of knowledge such as the business-analysis and software-testing syllabi widely used in the field. The contribution here is to apply them deliberately as data-quality controls in compliance and billing implementations, and to record the documents and sequence that make them effective.

3. Background: Data Quality Problems in Digital Compliance and Billing Systems

Compliance and billing systems are prone to data-quality problems for reasons that are structural rather than accidental. Their data often originates from several sources, including legacy systems, manual entry, and files supplied by other parties, each with its own conventions and gaps. Several teams touch the same records at different stages of a process, and responsibility for correctness is rarely assigned to any one of them. Deadlines for billing and compliance are fixed, so processing must proceed even when the underlying data is known to be imperfect.

Data quality is commonly described along dimensions such as completeness, validity, consistency, uniqueness, accuracy, and timeliness. A record can satisfy some of these and fail others: a customer record may be complete but duplicated, or a billing entry may be valid in format yet inconsistent with the service it purports to represent. Implementation defects usually involve one or more of these dimensions, and the control approach in this article is organised around detecting failures in each.

The central difficulty is that data defects are usually invisible at the point of creation. A missing compliance field or a billing record with no matching transaction does not announce itself; it surfaces weeks later during reconciliation or reporting, when investigation is required and the context is gone. Controls applied during requirements and testing move detection closer to creation, where the defect is cheapest to prevent.

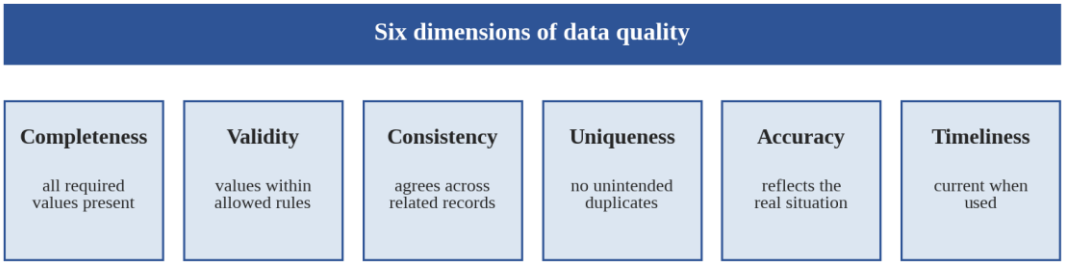

Figure 1. The six dimensions of data quality.

4. Purpose and Scope of the Article

The purpose of this article is to give implementation teams a clear method for using requirements traceability and acceptance testing to improve data quality, together with the supporting documents that make the method repeatable. It is written for business systems analysts, implementation and testing staff, and the compliance and finance reviewers who depend on the resulting reports.

The scope is limited to the implementation of digital compliance and billing systems and to the data those systems hold. It does not address software construction or infrastructure, and it does not propose specific tools. The controls described are method-level and can be applied with whatever requirements, testing, and tracking tools a project already uses. The emphasis throughout is on preventing and detecting data defects during implementation rather than correcting them after go-live.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

5. Digital Compliance and Billing Systems Covered

The method applies to the record types that carry the greatest data-quality risk in compliance and billing implementations. Five categories are addressed.

5.1 Customer and account records

The records that identify the customers or entities the system serves. Errors here, such as missing identifiers or duplicate accounts, propagate to billing, compliance, and reporting, which makes these records the highest-value point for review.

5.2 Billing and transaction records

The records of charges and the underlying transactions or service events that justify them. Their integrity requirement is that every charge is supported by a transaction and every chargeable transaction is billed once, with no omission and no duplication.

5.3 Compliance reporting fields

The fields a regulatory return requires, together with their permitted values. Completeness and validity of these fields determine whether a submission is accepted, so gaps create direct exposure for the operator and the entities it serves.

5.4 User approval records

The records of who reviewed or approved a transaction or correction, and on what authority. These records support the audit trail and allow a reviewer to confirm that changes were authorised.

5.5 Management reporting outputs

The reports produced for management and oversight. Their reliability depends on the records beneath them, so a reporting output is only as sound as the validation and reconciliation applied to its sources.

6. Common Data Quality Issues Observed in Implementation Projects

The issues below recur across compliance and billing implementations. They are presented as observed patterns so that the controls in later sections can be mapped to each.

Common implementation issue	Primary dimension affected
Missing mandatory fields	**Completeness**
Duplicate customer or account records	**Uniqueness**
Inconsistent account status	**Consistency**
Billing not matching transactions	**Consistency**
Unclear user approval steps	**Validity**
Incomplete exception logs	**Completeness**
Spreadsheet corrections outside system	**Accuracy**
Reports not matching source records	**Consistency**

Figure 2. Common implementation issues and the data-quality dimension each affects.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6.1 Missing mandatory fields

Records saved without fields that business or regulatory rules require, usually because the entry process permitted a save before all data was available. The gap stays hidden until a later process needs the field.

6.2 Duplicate customer or account records

The same customer or account represented more than once, often because no deterministic matching rule was applied at creation. Duplicates fragment billing and compliance history and produce reconciliation variances that are hard to trace.

6.3 Inconsistent account status

An account whose status is not consistent with its history or with related records, for example marked active while a required step remains incomplete. Inconsistent status produces reports that misstate the population served.

6.4 Billing records not matching service or transaction records

Charges with no matching transaction, or transactions that should be billed but are not. This mismatch is the most common source of revenue and billing exceptions.

6.5 Unclear user approval steps

Changes made without a clear record of who approved them and on what authority. Without this record, the operator cannot demonstrate that corrections were authorised.

6.6 Incomplete exception logs

Exceptions detected but not fully recorded, so they cannot be tracked to resolution. An incomplete log allows exceptions to be forgotten rather than closed.

6.7 Manual spreadsheet corrections outside the system

Corrections performed in spreadsheets rather than in the system, which removes the audit trail, hides the logic from review, and makes the result impossible to reproduce.

6.8 Reporting outputs not matching source records

Reports whose figures cannot be reconciled to the records they summarise, usually because the report logic and the source data have drifted apart. This is the failure most often noticed by management and reviewers.

7. Requirements Traceability as a Data Quality Control

Requirements traceability links each business requirement to the system requirements, test cases, and acceptance results that satisfy it. When applied as a data-quality control, it ensures that every rule governing the validity of a record is written down, implemented, and tested, and that no rule is lost between the business intent and the delivered system.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

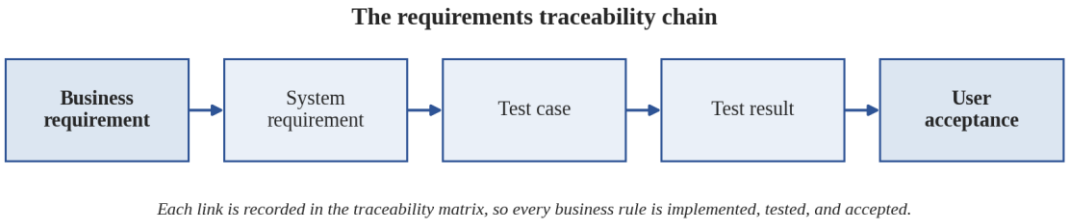

Figure 3. The requirements traceability chain.

7.1 Linking business requirements to system requirements

Each business rule that governs data, such as a mandatory field or a matching condition, is recorded as a business requirement and linked to the system requirement that implements it. The link makes it visible whether every rule has been carried into the system specification.

7.2 Linking system requirements to test cases

Each system requirement is linked to the test case that verifies it, so that no data rule is implemented without a test that confirms it works. This link is the basis for demonstrating coverage.

7.3 Linking test results to user acceptance

Each test result is linked to the acceptance decision it supports, so that signoff rests on evidence that the data rules were tested and passed rather than on assertion.

7.4 Tracking unresolved requirements before go-live

The matrix shows which requirements remain untested or failed, so that open data-quality risks are visible before go-live and can be accepted or resolved deliberately rather than discovered later.

7.5 Reducing ambiguity between business and technical teams

Because each requirement is stated once and traced through to a test, business and technical teams share a single definition of what correct data means, which removes the ambiguity that is a frequent source of defects.

8. Business Requirements Documentation

Traceability depends on requirements that are written clearly enough to be implemented and tested. The business requirements document is where the data rules are captured in business terms before they are translated into system behaviour.

8.1 Capturing business rules

The rules that determine whether a record is valid, such as which fields are required and which combinations are permitted, are recorded in plain business language so that they can be confirmed by the people who own them.

8.2 Defining data fields and expected values

Each data field is defined with its meaning, format, and permitted values, so that validation can be specified precisely and the same field is not interpreted differently by different teams.

8.3 Recording approval and review steps

The points at which a record or a change must be reviewed or approved are recorded, together with who is authorised to approve, so that approval becomes a defined control rather than an informal step.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

8.4 Identifying reporting requirements

The reports the system must produce are identified early, including the fields they draw on, so that the data needed for reporting is captured and validated rather than discovered to be missing at reporting time.

8.5 Documenting exception-handling rules

The rules for raising, classifying, and resolving exceptions are documented, so that exception handling is consistent and every exception has a defined path to closure.

9. Data Review During System Implementation

Data review during implementation confirms that records entering the system satisfy the rules defined in requirements. The checks below are applied as data is migrated or created, so that defects are caught before they reach production reporting.

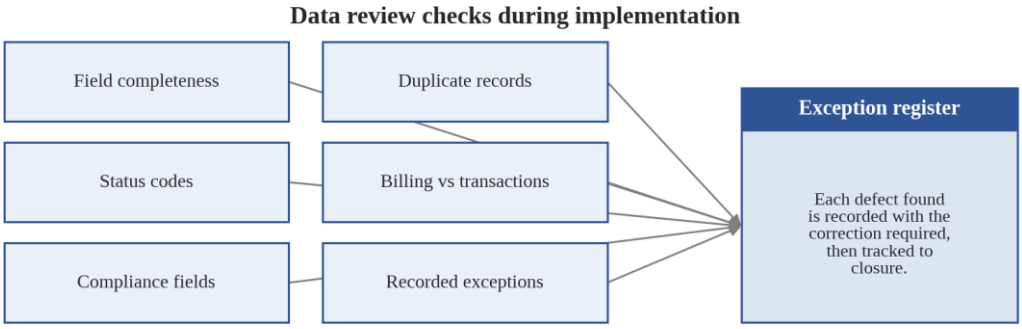

Figure 4. Data review checks during implementation, feeding the exception register.

9.1 Field completeness checks

Confirm that every field required by a business or compliance rule is present, catching the missing-field defect at its source.

9.2 Duplicate-record checks

Apply a matching rule to detect records that represent a customer or account already present, so duplicates are resolved before they fragment downstream history.

9.3 Status-code review

Confirm that each record's status is consistent with its history and with related records, so unreachable or contradictory states are corrected.

9.4 Cross-checking billing and transaction records

Confirm that each billing record has a matching transaction and each chargeable transaction has a billing record, detecting the mismatches that drive billing exceptions.

9.5 Reviewing compliance fields

Confirm that the fields required for regulatory submission are present and valid, so that returns are not rejected for incompleteness.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

9.6 Recording data exceptions for correction

Record each defect found during review in a single register, with the correction required, so that no defect is left untracked.

10. UAT Controls for Data Quality

User acceptance testing confirms that the system behaves correctly in the hands of the people who will use it. Designed as a data-quality control, acceptance testing exercises not only normal transactions but also the exception and reporting conditions where data defects appear. The practice draws on established software-testing principles applied to the data the system produces.

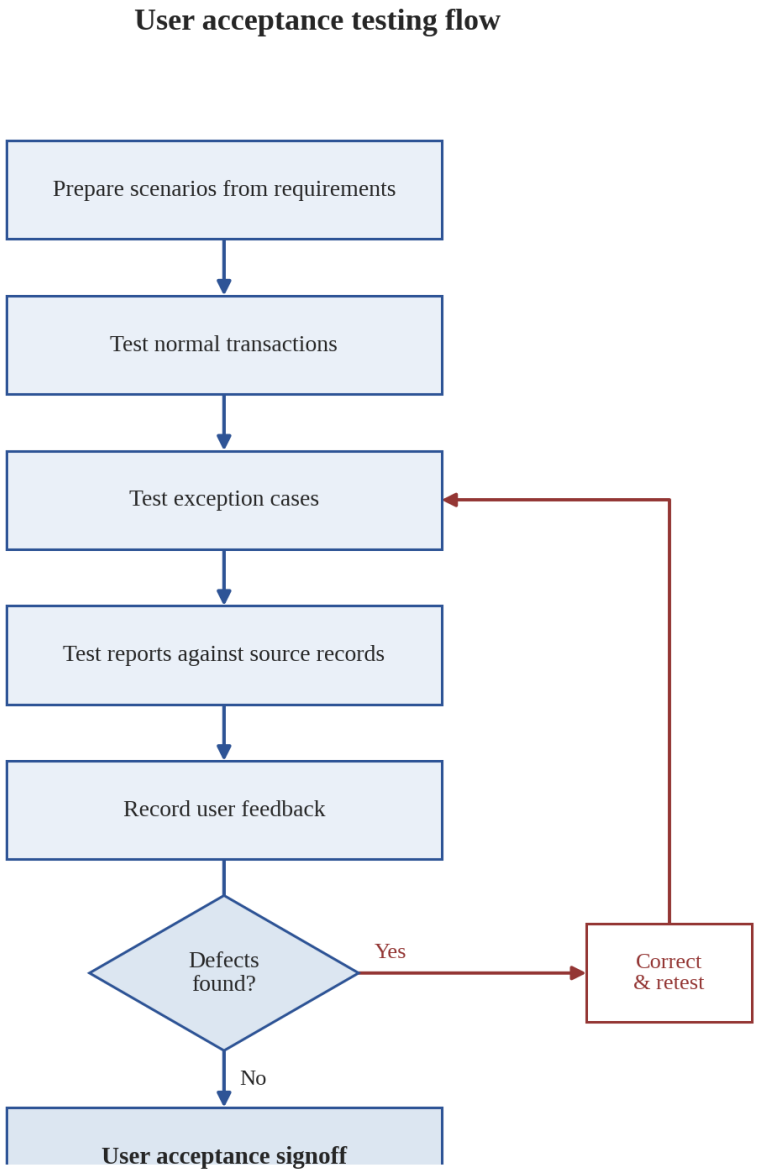

Figure 5. User acceptance testing flow, with the correction and retest loop.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

10.1 Preparing UAT scenarios from business requirements

Each scenario is derived from a business requirement and traced to it, so that testing confirms the data rules that requirements defined rather than an unrelated set of steps.

10.2 Testing normal user transactions

The common transactions are tested to confirm that records are created correctly and that validation accepts valid data without obstruction.

10.3 Testing exception cases

Invalid and exceptional data is entered deliberately to confirm that the system detects it and routes it correctly. Testing the failure paths is as important as testing the success paths, because that is where data defects originate.

10.4 Testing reports against source records

Reports are produced during testing and reconciled to their source records, confirming that reporting logic represents the underlying data rather than diverging from it.

10.5 Recording user feedback

The observations of the business users who run the tests are recorded, surfacing gaps between documented requirements and operational reality that earlier testing cannot.

10.6 Retesting corrected defects

Corrected defects are retested, and related cases are re-run, to confirm that a fix has resolved the defect without disturbing controls that previously passed.

10.7 Signoff before production release

Acceptance is recorded formally against defined exit criteria, with any residual defects acknowledged and their handling agreed, so that release is a documented decision.

11. Defect Logging and Issue Classification

A single defect log, with each item classified by type, allows the team to direct correction effort and to confirm that no item is carried into production unresolved. Classification matters because the right correction differs by type.

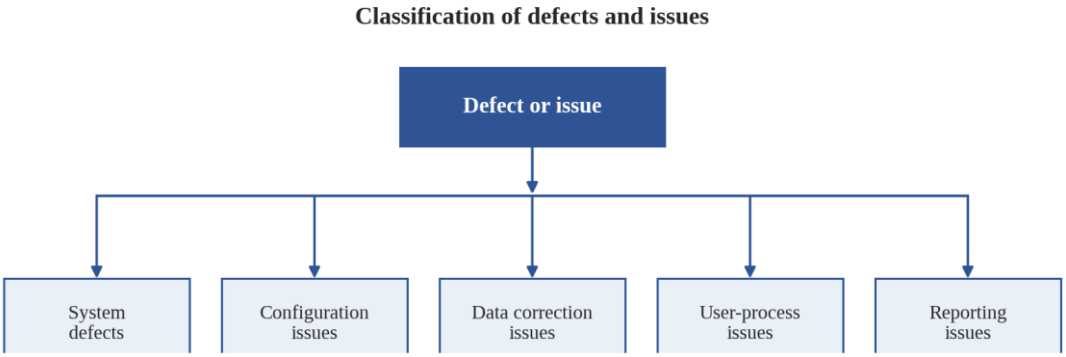

Figure 6. Classification of defects and issues.

11.1 System defects

Faults in how the system behaves, such as a validation rule that does not fire. These are corrected in the system and retested.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

11.2 Configuration issues

Defects arising from how the system is configured rather than how it is built, such as an incorrect permitted-value list. These are corrected in configuration and confirmed.

11.3 Data correction issues

Defects in the data itself, such as duplicate or incomplete records, that require the records to be corrected rather than the system to be changed.

11.4 User-process issues

Defects arising from how users operate the system rather than from the system, addressed through guidance or process change rather than code.

11.5 Reporting issues

Defects in reports, such as a figure that does not reconcile to source, corrected in reporting logic and confirmed against the source records.

11.6 Open-item review before go-live

All open items are reviewed before go-live so that each is either resolved or knowingly accepted, with no unresolved item passing silently into production.

12. Reconciliation Between Source Records and Reports

Reconciliation confirms that records which should agree do in fact agree, across systems and between reports and their sources. It detects defects that arise from drift between systems, which validation at entry cannot see.

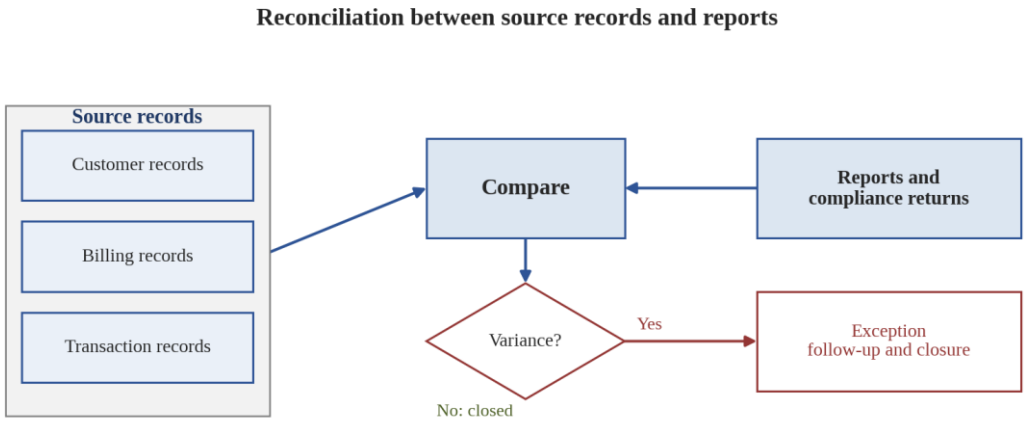

Figure 7. Reconciliation between source records and reports.

12.1 Customer record reconciliation

Compares customer records across the systems that hold them, confirming that each customer is represented consistently and that no system holds an orphaned or conflicting record.

12.2 Billing record reconciliation

Compares billed amounts against the records that support them, confirming that every charge is justified and every justified charge is billed.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

12.3 Transaction record reconciliation

Compares transaction records against the billing and service records that should correspond to them, detecting transactions that were billed incorrectly or not at all.

12.4 Compliance report reconciliation

Compares the figures in a compliance return against the source records, confirming that the return faithfully represents the data it reports.

12.5 Exception follow-up and closure

Tracks each variance to resolution, confirming that variances are corrected and closed rather than carried forward into the next cycle.

13. Management Reporting and Implementation Visibility

Implementation visibility comes from reporting the state of the controls themselves, so that those accountable can see open risks and act before go-live. The items below give management a clear view of readiness.

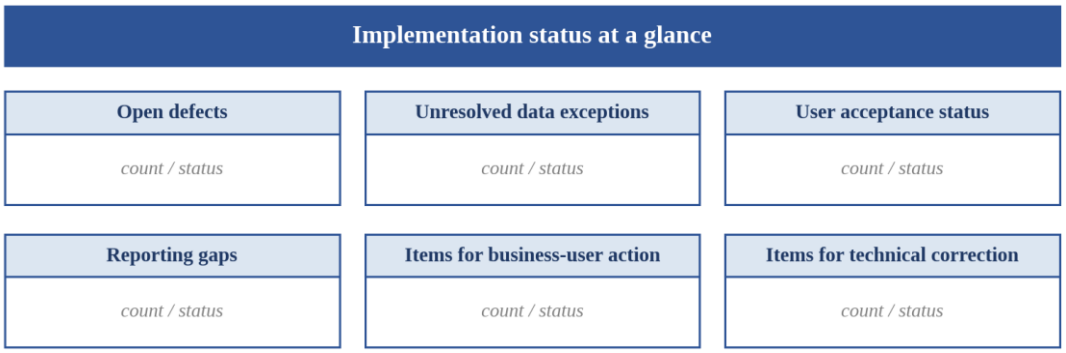

Figure 8. Implementation status at a glance.

13.1 Open defects

The current population of unresolved defects by type and severity, so that correction effort is directed and readiness can be judged.

13.2 Unresolved data exceptions

The exceptions recorded during data review that remain open, showing where the data is not yet ready to support reporting.

13.3 User acceptance status

The proportion of acceptance scenarios passed, failed, and outstanding, traced to requirements, so that acceptance is measured rather than assumed.

13.4 Reporting gaps

The reports that do not yet reconcile to source, so that reporting defects are visible before management relies on the figures.

13.5 Items requiring business-user action

The open items whose resolution depends on a business user, such as a rule clarification or a data decision, so that ownership is clear.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

13.6 Items requiring technical correction

The open items whose resolution depends on the technical team, so that the two streams of work are tracked distinctly and neither stalls for want of the other.

14. Recommended Implementation Control Documents

The documents below record the controls, the evidence that they were applied, and the decisions taken. Together they allow the implementation to be reviewed and the data to be relied upon.

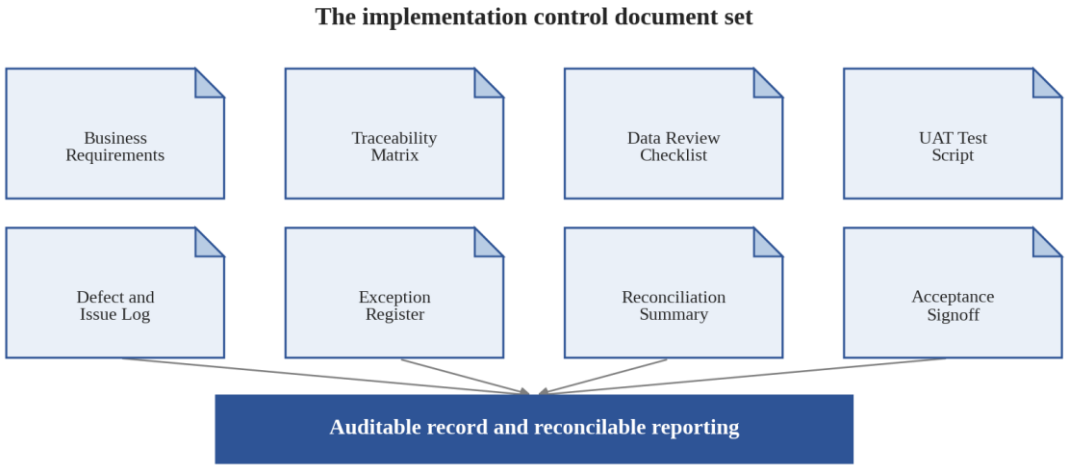

Figure 9. The implementation control document set.

14.1 Business Requirements Document

States the business rules and data requirements the system must satisfy, in business language confirmed by their owners.

14.2 Requirements Traceability Matrix

Links each requirement to the system requirement, test case, and acceptance result that satisfy it. A sample is provided in Appendix A.

14.3 Data Review Checklist

Lists the data-review checks and records their results during implementation. A sample is provided in Appendix B.

14.4 UAT Test Script

Provides the steps and expected results for acceptance scenarios, including exception and reporting cases. A sample is provided in Appendix C.

14.5 Defect and Issue Log

Records defects from detection to resolution, classified by type. A sample is provided in Appendix D.

14.6 Exception Register

Records data exceptions found during review and their correction, so that each is tracked to closure.

14.7 Reconciliation Summary

Records the reconciliations performed and the variances found and resolved. A sample is provided in Appendix E.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

14.8 User Acceptance Signoff Form

Records formal acceptance against exit criteria, with residual defects acknowledged. A sample is provided in Appendix F.

15. Practical Implementation Model

The controls are applied in a sequence in which each step depends on the one before it. The sequence keeps data quality a continuous concern from requirements through to signoff.

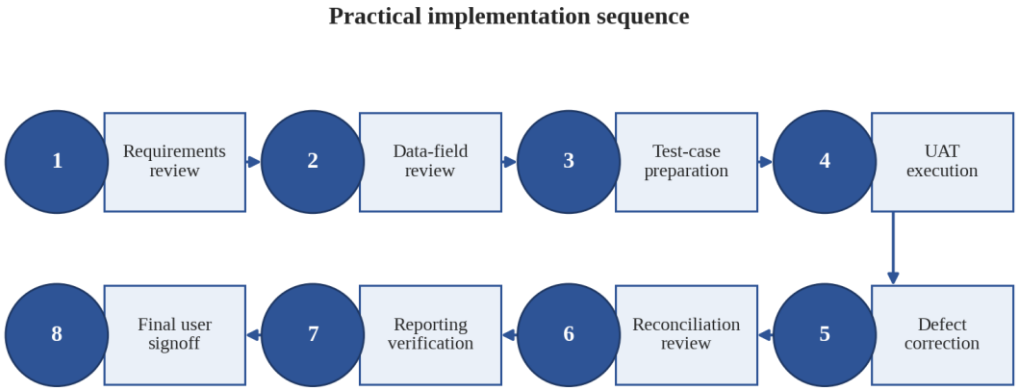

Figure 10. The practical implementation sequence.

15.1 Requirements review

Confirm that the business rules and data requirements are complete and agreed, and record them in a form that can be traced and tested.

15.2 Data-field review

Confirm that every data field is defined with its format and permitted values, so that validation can be specified precisely.

15.3 Test-case preparation

Prepare test cases from the requirements, including exception and reporting cases, and trace each to the requirement it verifies.

15.4 UAT execution

Run the acceptance scenarios with business users, exercising normal, exception, and reporting conditions, and record the results.

15.5 Defect correction

Correct and reclassify defects, then retest, so that each is resolved or knowingly accepted before go-live.

15.6 Reconciliation review

Reconcile records across systems and reports to their sources, and resolve variances.

15.7 Reporting verification

Confirm that each report reconciles to its source records and meets its stated requirement.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

15.8 Final user signoff

Record formal acceptance against the exit criteria, completing the sequence with a documented readiness decision.

16. Benefits of the Traceability and UAT Control Approach

The approach is justified by the benefits it produces during and after implementation, each following from a control applied at the right stage.

16.1 Improved data completeness

Field and review controls catch missing data at creation, so records reach production complete rather than gapped.

16.2 Earlier identification of billing errors

Billing-to-transaction checks and reconciliation surface billing errors during implementation, before they reach customers or revenue figures.

16.3 Better classification of system and data issues

Defect classification distinguishes system, configuration, data, process, and reporting issues, so each is corrected in the right way.

16.4 Reduced manual correction after go-live

Defects prevented or caught before go-live reduce the manual correction that would otherwise be needed in production, where it is most disruptive.

16.5 Improved coordination between business and technical teams

Traceability gives both teams a single definition of correct data, which reduces the rework caused by differing interpretations.

16.6 More reliable compliance and management reporting

Reports built on validated and reconciled records can be relied upon and traced to their sources.

17. Limitations

The approach has limits that should be stated plainly. Its effectiveness depends on the quality of the requirements: rules that are not captured cannot be traced or tested, so a control is only as good as the requirement behind it. Traceability and acceptance testing add effort during implementation, and on a project under severe time pressure that effort competes with delivery deadlines; the value is realised later, in reduced correction, which is not always visible when the investment is made.

The controls detect and prevent defects, but they do not by themselves repair poor source data; where data arrives defective from a legacy system or a third party, correction still requires effort beyond the controls described here. The approach also depends on people: review checks, defect classification, and signoff are performed by staff, and their judgement and diligence affect the result. Finally, the method assumes that business and technical teams are willing to agree a single definition of correct data; where that agreement is not reached, traceability records the disagreement but cannot resolve it.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

18. Conclusion

Data quality in compliance and billing systems is decided during implementation, not after it. The defects that surface during reporting are usually the visible result of rules that were never written down clearly and behaviours that were never tested against realistic data. Requirements traceability and user acceptance testing address both causes: traceability ensures that every data rule is captured, implemented, and tested, and acceptance testing confirms that the system enforces those rules under the normal, exception, and reporting conditions where defects appear.

Applied together, with the supporting documents and the sequence described here, these controls move defect detection close to its source, where correction is cheapest, and produce compliance and management reports that can be reconciled to the records beneath them. The method is ordinary in its parts; its value comes from applying the parts deliberately and recording the evidence that they were applied. Used consistently, it allows an implementation team to hand over a system whose data can be relied upon from the first reporting cycle.

19. References and Sources Consulted

The methods described in this article draw on recognised standards and professional practice guides. The principal sources consulted are listed below.

[1] International Institute of Business Analysis. A Guide to the Business Analysis Body of Knowledge (BABOK Guide), Version 3. Toronto: International Institute of Business Analysis, 2015.
[2] International Software Testing Qualifications Board. Certified Tester Foundation Level Syllabus. Version 2018.
[3] ISO/IEC/IEEE. ISO/IEC/IEEE 29148:2018 — Systems and Software Engineering — Life Cycle Processes — Requirements Engineering. Geneva: International Organization for Standardization, 2018.
[4] ISO/IEC. ISO/IEC 25012:2008 — Software Engineering — Software Product Quality Requirements and Evaluation (SQuaRE) — Data Quality Model. Geneva: International Organization for Standardization, 2008.
[5] ISO. ISO 9001:2015 — Quality Management Systems — Requirements. Geneva: International Organization for Standardization, 2015.
[6] DAMA International. DAMA-DMBOK: Data Management Body of Knowledge. 2nd ed. Basking Ridge, NJ: Technics Publications, 2017.
[7] Institute of Electrical and Electronics Engineers. IEEE Standard for Software and System Test Documentation. IEEE Std 829-2008. New York: IEEE, 2008.
[8] International Accounting Standards Board. Conceptual Framework for Financial Reporting. London: IFRS Foundation, 2018.

20. Appendices and Templates

The templates below illustrate the control documents described in this article and can be adapted to a specific implementation.

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Appendix A: Sample Requirements Traceability Matrix

A simplified extract linking each requirement to the test and acceptance result that satisfy it.

Req ID	Business Rule	System Requirement	Test Case	UAT Result
BR-01	Mandatory fields required at save	Reject save if mandatory field empty	TC-007	Passed
BR-02	One record per customer	Match on customer key at creation	TC-012	Passed
BR-03	Charge requires matching transaction	Validate billing against transaction	TC-019	Passed
BR-04	Changes require recorded approval	Capture approver and authority	TC-024	Passed
BR-05	Report total matches source	Reconcile report to source records	TC-031	In test

Appendix B: Sample Data Review Checklist

Ref	Check	Applied To	Result
DR-01	All mandatory fields present	Customer and account records	Pass
DR-02	No duplicates on customer key	Customer records	Pass
DR-03	Status consistent with history	Account records	Pass
DR-04	Each charge has a transaction	Billing records	Pass
DR-05	Compliance fields complete	Compliance reporting fields	Pass
DR-06	Exceptions recorded for correction	Exception register	Pass

Appendix C: Sample UAT Test Script

Scenario: create a customer account and raise a charge, confirming that data rules are enforced.

Step	Action	Expected Result	Result
1	Save account with a mandatory field empty	Save is rejected with a clear message	Pass
2	Create account matching an existing key	Duplicate is detected and blocked	Pass
3	Raise a charge with no transaction	Charge is rejected as unmatched	Pass
4	Change a record without approval	Change is blocked pending approval	Pass

Journal of Information Systems Engineering and Management
2021, 6(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Step	Action	Expected Result	Result
5	Produce the account report	Report total reconciles to source	Pass

Appendix D: Sample Defect and Issue Log

ID	Description	Type	Severity	Status
D-01	Empty mandatory field accepted	System	High	Fixed
D-02	Permitted-value list incorrect	Configuration	Medium	Fixed
D-03	Duplicate customer records found	Data	High	Corrected
D-04	Users skipping approval step	User-process	Medium	Guidance issued
D-05	Report total not reconciling	Reporting	High	In progress

Appendix E: Sample Reconciliation Summary

Area	Source	Report	Variance	Action
Customer records	12,480	12,480	0	Closed
Billing records	9,212	9,205	7	Investigating
Transactions	9,212	9,212	0	Closed
Compliance return	Matches	Matches	0	Closed

Appendix F: Sample User Acceptance Signoff Form

Item	Detail
System	Digital compliance and billing system
Release	Implementation release for go-live
UAT period	Start date to end date
Scenarios passed	Count of total
Open defects accepted	List with agreed handling
Conditions for release	Any conditions attached to acceptance
Business owner signoff	Name, role, signature
Date	Date of acceptance