Barry McManus

Hugh O'Neill

IT VENDOR AUDITS: A PARADIGM SHIFT FROM DOCUMENTATION QUALITY TO PRODUCT QUALITY

This series of articles has explored a fundamental challenge facing the pharmaceutical industry: how to assess vendor software product quality beyond the examination of documentation.

Throughout this series, we have challenged a prevalent assumption in the regulated industry: that documented processes equate to quality. As McDowall stated, "The assumption is that a documented process equates to quality, but this is a fallacy."[1]

Our vendor audit travels have revealed a consistent pattern:

- Quality functions measure documentation compliance, not software product quality
- Vendor QMS documents describe what should be done, but not how to do it
- Internal audits focus on validation artifacts rather than engineering practices
- Testing efforts concentrate on proving requirements work, not on uncovering defects.

The consequence? Software products with high documentation quality but low operational quality, leading to defects in production, data integrity risks, regulatory delays and significant remediation costs for the regulatory industry.

This final instalment summarises some of the insights from Parts One through Four to allow for referencing by those conducting vendor assessments.

'Throughout this series, we have challenged a prevalent assumption in the regulated industry: that documented processes equate to quality.'

ARTICLE 1: QUASAR #170[2]: THE QUALITY PIVOT FOR THE IT AUDITOR

- Testing is a finite activity, requiring a risk-based focus on the technical, as well as regulatory risks
- A test strategy involves different test phases, where each test phase is suited to identifying specific software product defects
- Vendor over-focus on acceptance end-to-end testing driven by perceived auditor expectation is conducted at the cost to unit, integration, system test phases
- Vendor audit samples demonstrated:
  - A QMS focussed on the what is done and not the how
  - Internal audit focused on documentation quality over software product quality
  - Testing focused on 'validation' evidence generation over software product quality level measurement
- Software product quality was defined as the absence of defects that cause failures or incorrect data, which enabled quantitative measurement
- Defect metrics facilitate trending that facilitates objective assessment of software product quality and associated QMS process maturity.

FIGURE 1. PROBLEM OF AN ACCEPTANCE LEVEL ONLY TEST FOCUS

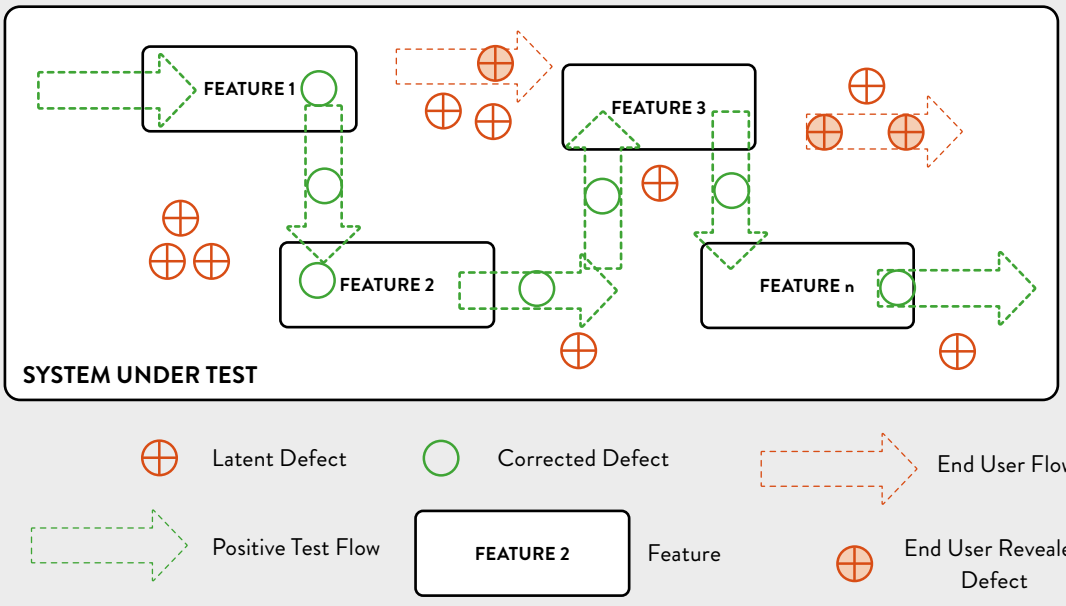


FIGURE 2. TESTING STRATEGY OVERVIEW EXAMPLE

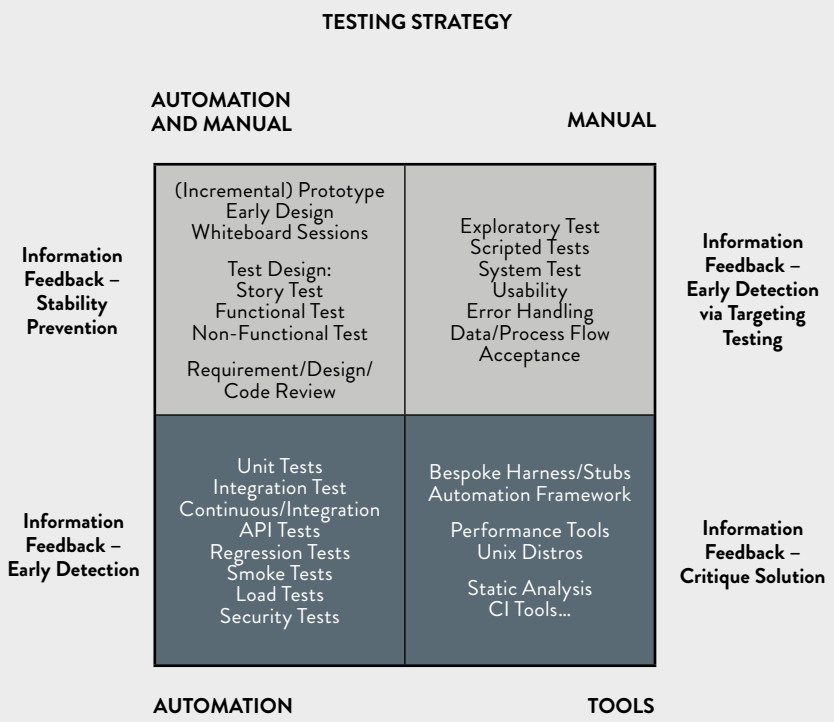
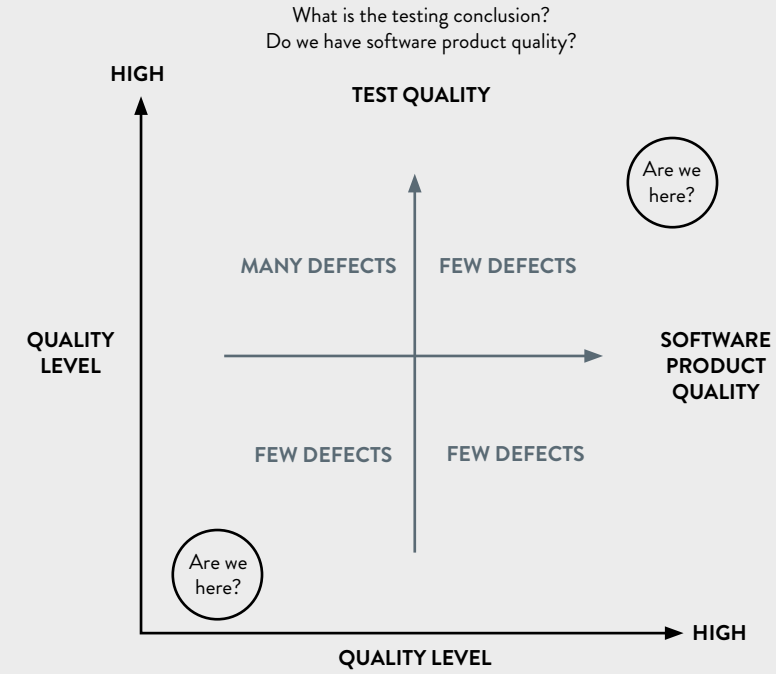


FIGURE 3. WHAT DO VERIFICATION ACTIVITIES TELL US ABOUT SOFTWARE PRODUCT QUALITY?



ARTICLE 2: QUASAR #171[3]: TEST STRATEGIES AND TECHNIQUES FOR THE IT AUDITOR

- Better testing strategies lead to better defect discovery and higher levels of software product quality
- The more varied the testing techniques applied, the higher the likelihood of detecting defects before software reaches operational use. Individual testing techniques have relatively low defect detection efficiency, requiring multiple approaches
- Testing strategies discussed:
  - The effectiveness of static testing (walkthroughs, reviews) at detecting defects
  - Dynamic testing of white and black box testing techniques to challenge the software product under execution
  - Examples of test techniques within these domains
- Each test activity affords measurements that can be utilised in assessment of the software product quality levels and associated software product line (QMS) processes.

ARTICLE 3: QUASAR #172[4]: SOFTWARE QUALITY ANALYTICS FOR THE IT AUDITOR

If a vendor focuses on documentation style over actual software product quality, the repetition of informal acceptance-level tests to produce compliant documentation is of limited value to the software product quality levels.

- The vendor QMS often fails to reflect actual production line processes because:
  - The QMS describes the 'what', neglecting the 'how' to perform technical process steps
  - Internal audit focused on validation artifacts rather than technical process adherence and effectiveness
  - CAPA process that doesn't address technical software lifecycle deficiencies
  - Regulated customer reliance on vendor testing may be misplaced
- The software lifecycle is a production line requiring verification at each phase to eliminate defect proliferation in subsequent production line phases and in operational use
- Leveraging the techniques discussed in article two
- Introduced Software Quality Analytics to assess the software product quality level

- Provided three real world examples of the use of Software Quality Analytics:
  - Static verification at requirements phases preventing 53 defects from being built into a system
  - Using defect metrics analysis as a quality gate and to predict software product quality issues in operational use
  - Using active software quality analytics to demonstrate QMS improvements across a 30 month period (shorter than the follow up audit cycle)
- Using Software Quality Analytics to compare software vendors quality maturity.

Software product measurement throughout the production process is fundamental to quality management. Vendors with a mature QMS actively measure process outputs, track defect trends, predict quality levels and use analytics to drive continuous improvement – not just produce compliant documentation.

ARTICLE 4: QUASAR #173[5]: TEST AUTOMATION FOR THE IT AUDITOR

Automation increases testing speed and coverage but requires significant investment in strategy, design and maintenance.

- Strengths:
  - Regression and smoke testing
  - API and integration testing
  - Complex technical testing such as performance, load and stress testing
  - Unit (code) testing
- Limitations:
  - Not a replacement for human critical thinking
  - Maintenance overhead
  - Need to understand the scope of the automation in order to assess its effectiveness
  - Proliferation of redundant and non-value added automated tests can provide false confidence
- Success factors:
  - Strategy
  - Lifecycle management
  - Self-diagnosis
  - Technical skills
  - Human in the loop – combination with manual testing
- Common approaches:
  - Record/playback: fast but fragile
  - Data driven: test data is separate from automation scripts: flexible, robust and maintainable
  - Key word driven: readable and usable by non-technical staff
  - Hybrid: bespoke combination of the previous two approaches.

Success depends on selecting the right tests to automate, proper lifecycle management and integrating with – not replacing – manual testing efforts.

FIGURE 4. TEST AUTOMATION MANAGEMENT CONSIDERATIONS

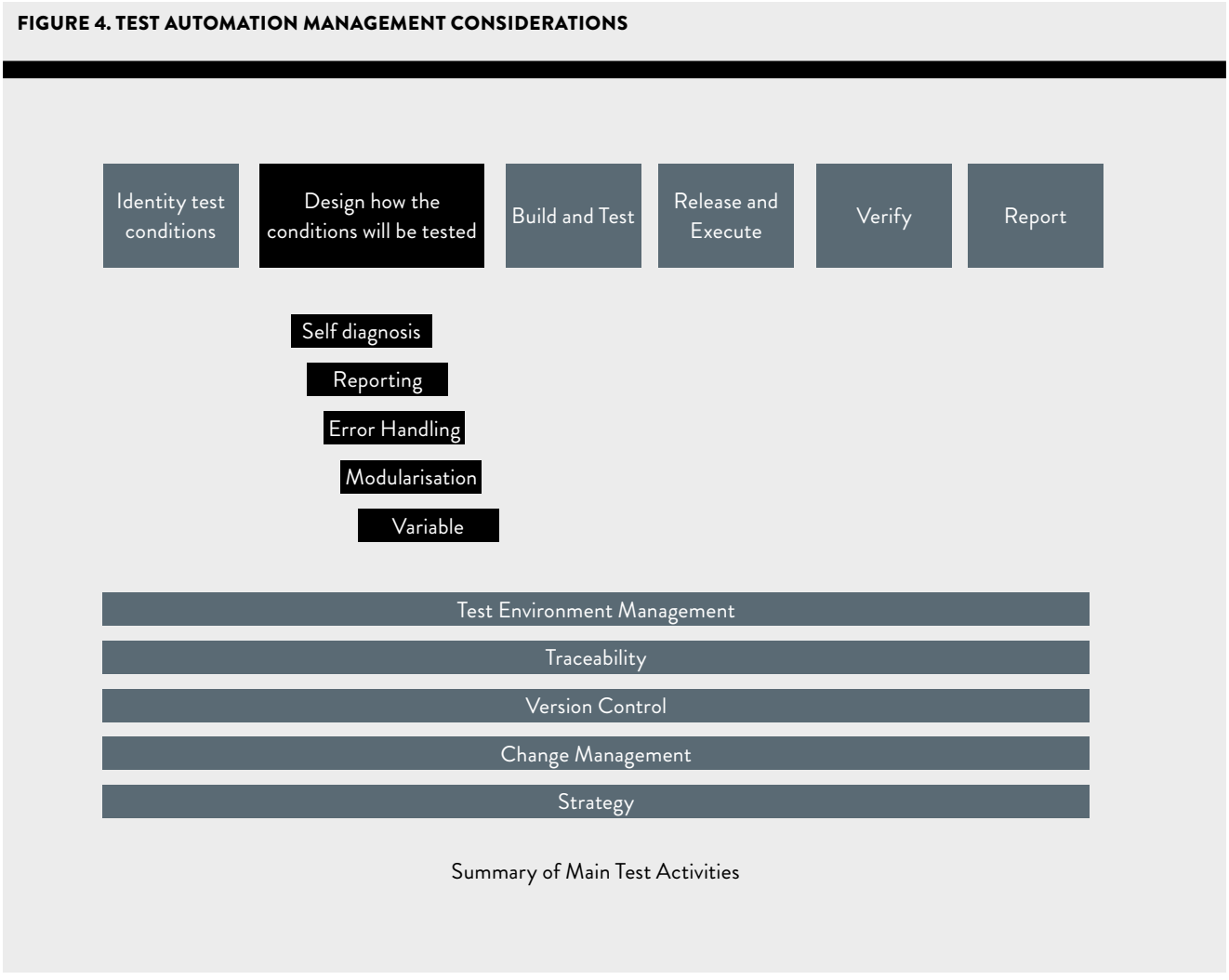
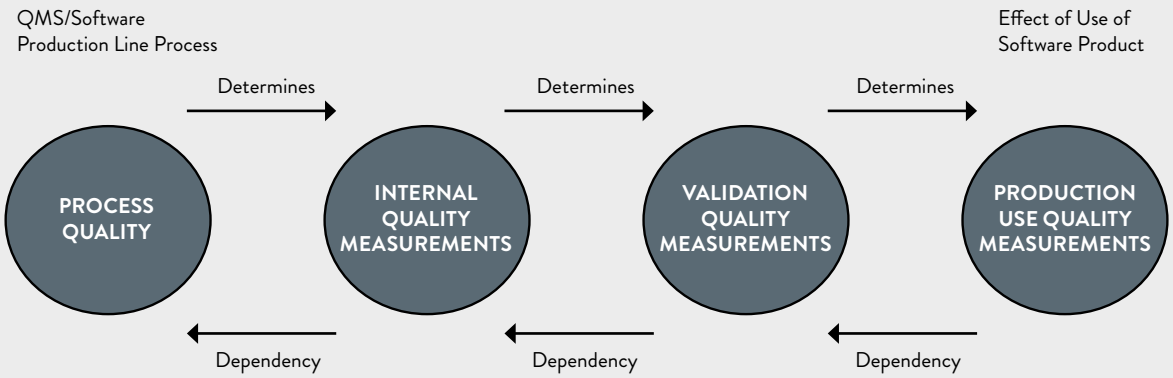


FIGURE 5. ISO91261: RELATIONSHIP BETWEEN THE SLC QUALITY PROCESS AND SOFTWARE PRODUCT QUALITY MEASUREMENT



These articles challenged the assumption that documentation quality equals software product quality and calls for a pivot toward measuring software product quality through testing strategy, coverage and defect trends.

'When you can measure what you are speaking about, and express it in numbers, you know something about it.' – Lord Kelvin[6]

FOR QUALITY ASSURANCE

- Documentation quality ≠ software quality
- Focus QA oversight on **test strategy, test coverage and defect trends**, not just validation artefacts
- Early-stage testing (static specification reviews, unit, integration, system) is critical to preventing defects that later impact data integrity and patient safety
- Use **defect metrics** as objective evidence of vendor software product quality
- Risk-based testing aligns with FDA CSA's 'least burdensome' intent.

FOR IT/SOFTWARE ENGINEERING

- Static and dynamic testing must aim to **prevent and early detect defects**, not only prove requirements are met
- Strong software quality depends on:
  - Effective unit and integration testing
  - Independent system testing
  - Clear separation between testing and debugging
- Static testing (reviews, design checks) is as important as dynamic testing
- Falling defect trends across releases indicate a **maturing software production line process**.

FOR AUDITORS

- Do not equate extensive test documentation with effective testing
- Assess **how testing is performed**, not just what documents exist
- Look for evidence of:
  - Multi-level testing (unit→acceptance) across the software production line
  - Independent testing activities
  - Defect measurement and trend analysis, not just documentation compliance
  - Whether test strategy includes prevention (reviews, inspections) not just detection (testing)
- Ask how vendors measure and improve software product quality, not how they generate validation artefacts.

REFERENCES

1. Bob McDowall, Computer Software Assurance: Perfect Solution or Confidence Trick? Technology Networks, 15 November, 2024,
2. Barry McManus & Hugh Oneil, Software Testing, Measuring Vendor Software Quality – Part One, Quasar 170, February 2025
3. Barry McManus & Hugh Oneil, Software Testing, Measuring Vendor Software Quality – Part Two, Test Strategies and Techniques, Quasar 171, May 2025
4. Barry McManus & Hugh Oneil, Software Testing, Measuring Vendor Software Quality – Part Three, Software Quality Analytics, Quasar 172, August 2025
5. Barry McManus & Hugh Oneil, Automation Testing, Quasar 173, November 2025
6. Kelvin, Electrical Units of Measurement, Kelvin, PLA, Vol 1, 1883

PROFILES

Barry is a Principal Consultant for Empowerment Quality Engineering Ltd, a Computerised System Regulatory consultancy that bridges the gap between IT and quality.

He focuses on building quality and security into Computerised Systems (CS) by using quality techniques from the wider software industry while ensuring regulatory compliance. He leads GxP CSV compliance and IT Supplier/Service Provider audits across the globe; performs IT supplier's software life cycle process improvement, risk assessments to drive validation strategies, validation projects and tailored training.

Barry has over 27 years' experience in Quality Assurance, Software Engineering and IT Administration with vast technical knowledge of every role and every activity within the CS life cycle; including multiple technologies, development methodologies (traditional and agile), databases and programming languages.

He is a member of the RQA IT Committee, the MARSQA and was a member of the ISPE Data Integrity Project team.

Hugh is VP Operations and Quality at PHARMASEAL International Ltd and an independent computer systems validation consultant.

He is an IT professional with over 35 years of experience of using technology in the pharmaceutical industry, initially as a developer, later an implementer and more recently specialising in compliance.