LLM Evaluation Framework for S1000D-Conformant IETM Development

Design Science Research Study

⚠ DISCLAIMER — PORTFOLIO WRITING SAMPLE

This document is a technical writing sample created solely to demonstrate the authoring skills and domain expertise of Nathan B. Smith. All organizations, programs, project data, and research findings presented herein are entirely hypothetical and were created for illustrative purposes only. This document does not contain any proprietary, confidential, classified, controlled unclassified (CUI), or export-controlled information from any current or former employer, client, or government program. No real program data, trade secrets, or intellectual property belonging to any organization has been used in the preparation of this sample.

This work is intended exclusively for professional portfolio use and is not affiliated with, endorsed by, or representative of any defense contractor, government agency, or commercial entity.

© COPYRIGHT NOTICE

Contact: nbsmith@outlook.com

Document No: NBS-DSR-2024-001 **Revision:** A **Date:** June 2024 **Classification:** UNCLASSIFIED // PORTFOLIO SAMPLE DOCUMENT

Author: Nathan B. Smith, DCS Doctor of Computer Science — Big Data Analytics Colorado Technical University

Distribution Statement A — Approved for public release; distribution unlimited.

Document Control

Revision History

Rev	Date	Author	Description
—	2024-01-15	N. Smith	Initial draft
A	2024-06-01	N. Smith	Final revision incorporating committee feedback

Applicable Documents

ID	Title	Issue/Rev
S1000D	International Specification for Technical Publications	Issue 5.0
ASD-STE100	Simplified Technical English Specification	Issue 8
MIL-STD-40051-2	Preparing Digital Technical Information for IETM	Rev B
MIL-STD-3031	DoD Standard Practice for Technical Data Packages	Rev A
DO-178C	Software Considerations in Airborne Systems	—

Abstract

This design science research study presents an empirical evaluation framework for assessing the application of Large Language Models (LLMs) in the development of S1000D-conformant Interactive Electronic Technical Manuals (IETMs) for Department of Defense (DoD) and Department of the Workforce (DoW) programs. The study addresses a critical gap in the existing literature regarding the systematic evaluation of generative AI technologies for structured technical authoring in defense and aerospace contexts.

The framework encompasses three evaluation dimensions: **content accuracy** (semantic correctness, terminology compliance, and technical precision), **structural conformance** (S1000D schema validation, BREX rule compliance, and data module code integrity), and **operational efficiency** (authoring throughput, review cycle reduction, and cost-per-module metrics). Experimental trials were conducted using GPT-4, Claude, and Llama 2 models against a baseline of expert human-authored data modules across descriptive, procedural, and fault isolation content types.

Results demonstrate that LLM-assisted authoring pipelines achieve 87.3% first-pass schema validation rates and reduce average authoring time by 62% compared to manual workflows, while maintaining content accuracy scores within 4.2% of human expert baselines. The study concludes with a validated artifact — a Python-based evaluation pipeline — and actionable recommendations for integrating LLM technologies into S1000D IETM development programs.

Keywords: Large Language Models, S1000D, IETM, Design Science Research, natural language processing, technical authoring, defense documentation, XML validation, evaluation framework, artificial intelligence

---

Table of Contents

Glossary of Key Terms

Term	Definition
BREX	Business Rules Exchange — mechanism for defining project-specific S1000D business rules
CSDB	Common Source Database — repository for S1000D data modules
DMC	Data Module Code — unique identifier for each S1000D data module
DSR	Design Science Research — research paradigm for creating and evaluating IT artifacts
IETM	Interactive Electronic Technical Manual
LLM	Large Language Model — neural network trained on large text corpora
SNL	Structured Name List — controlled terminology list in S1000D
STE	Simplified Technical English — controlled language specification for aerospace
TDP	Technical Data Package — collection of source engineering data

---

Chapter 1 — Introduction

1.1 Background and Context

The development of Interactive Electronic Technical Manuals (IETMs) conforming to the S1000D international specification represents one of the most resource-intensive documentation activities in defense and aerospace programs. Each data module requires precise adherence to XML schema definitions, Business Rules Exchange (BREX) compliance, controlled terminology from approved Structured Name Lists (SNLs), and rigorous quality assurance workflows that typically involve multiple subject matter expert (SME) review cycles.

The S1000D specification, currently at Issue 5.0, defines over 80 distinct data module schema types, each with specific structural requirements, mandatory elements, and conditional content rules. A typical defense acquisition program may require development of 500 to 5,000+ data modules, with each module requiring an average of 8–16 hours of authoring, review, and quality assurance effort (Smith & Johnson, 2023). The cumulative cost of IETM development on major weapons system programs frequently exceeds $10M over the system lifecycle.

Recent advances in Large Language Model (LLM) technology — including GPT-4 (OpenAI, 2023), Claude (Anthropic, 2024), and open-source alternatives such as Llama 2 (Meta, 2023) — have demonstrated significant capabilities in natural language generation, structured data transformation, and domain-specific content production. These capabilities suggest potential for substantial efficiency gains in technical authoring workflows. However, the application of LLM technologies to standards-conformant technical publications in regulated defense environments has not been systematically evaluated.

1.2 Problem Statement

Defense acquisition programs face increasing pressure to reduce technical data development costs while maintaining compliance with MIL-STD-40051-2 and S1000D requirements. Current IETM authoring workflows are predominantly manual, requiring specialized XML editors (e.g., Oxygen XML, Arbortext), deep knowledge of S1000D schema structures, and iterative SME review cycles that introduce significant schedule risk.

While commercial AI writing tools have proliferated, none have been empirically validated for use in S1000D-conformant content development. Program managers and contracting officers lack evidence-based guidance for evaluating whether and how LLM technologies can be integrated into existing IETM development workflows without compromising data quality, schema conformance, or regulatory compliance.

1.3 Research Questions

This study addresses three primary research questions:

RQ1: To what extent can LLM-generated content meet S1000D schema validation and BREX compliance requirements without human intervention?

RQ2: How does the technical accuracy of LLM-generated data module content compare to human expert-authored baselines across descriptive, procedural, and fault isolation information types?

RQ3: What is the cost-benefit profile of LLM-assisted authoring pipelines versus traditional manual workflows for S1000D IETM development programs?

1.4 Scope and Limitations

This study evaluates LLM performance specifically within the context of S1000D Issue 5.0 data module development for DoD programs. The evaluation scope encompasses three data module information types: descriptive (DM type descript), procedural (DM type proced), and fault isolation (DM type fault). The study does not evaluate LLM performance for illustrated parts data (IPD), wiring data, or crew/operator content types, which are identified as areas for future research.

> **NOTE:** All data modules used in this study were developed using unclassified technical content. No classified, controlled unclassified information (CUI), or export-controlled data was used in LLM training, prompting, or evaluation.

Chapter 2 — Methodology

2.1 Design Science Research Framework

This study employs the Design Science Research (DSR) methodology as codified by Hevner et al. (2004) and refined by Peffers et al. (2007). DSR is particularly suited to this research because it explicitly addresses the creation and evaluation of IT artifacts — in this case, an LLM-based evaluation framework — within a defined problem context. The methodology provides rigorous guidelines for ensuring that the research artifact is both practically useful and theoretically grounded.

The DSR process model consists of six activities applied iteratively: (1) Problem Identification and Motivation, (2) Definition of Objectives for a Solution, (3) Design and Development, (4) Demonstration, (5) Evaluation, and (6) Communication. Each activity produces specific outputs that inform subsequent phases, with evaluation results feeding back into the design cycle for artifact refinement.

2.1.1 DSR Activity Mapping

DSR Activity	Research Output	Methods Applied
1. Problem Identification	Literature gap analysis; stakeholder needs assessment	Systematic literature review; SME interviews (n=12)
2. Objectives Definition	Evaluation criteria; performance thresholds	Requirements analysis; benchmark definition
3. Design & Development	Python evaluation pipeline; prompt engineering templates	Agile development; iterative prototyping

DSR Activity	Research Output	Methods Applied
4. Demonstration	Pilot evaluation on 50 data modules	Controlled experiment; baseline comparison
5. Evaluation	Statistical analysis of accuracy, conformance, efficiency	Paired t-tests; ANOVA; effect size (Cohen's d)
6. Communication	Dissertation; journal submission; this document	Academic writing; technical reporting

Table 2-1. DSR Activity Mapping to Research Outputs

2.2 Experimental Design

The experimental design follows a between-subjects factorial structure with two independent variables: (1) **authoring method** (four levels: human expert, GPT-4 assisted, Claude assisted, and Llama 2 assisted) and (2) **content type** (three levels: descriptive, procedural, and fault isolation). The dependent variables are content accuracy score, schema validation pass rate, and authoring time-to-completion.

A total of 200 data modules were produced: 50 per authoring method, distributed evenly across the three content types (with additional descriptive modules to balance the design). Human expert baselines were authored by three S1000D-certified technical writers with an average of 12 years of IETM development experience.

2.2.1 Experimental Controls

The following controls were applied to minimize confounding variables:

Source material: All authoring methods received identical source technical data packages (TDPs) with the same engineering drawings, specifications, and SME-approved content outlines.

Schema version: All data modules were validated against S1000D Issue 5.0 schemas with identical BREX rules (project-specific Business Rules Decision Point selections).

Terminology: A common Structured Name List (SNL) derived from ASD-STE100 Issue 8 was enforced across all authoring conditions.

Evaluation blind: All data modules were anonymized prior to quality review to prevent evaluator bias toward any authoring method.

2.3 Evaluation Metrics Definition

The evaluation framework defines 14 discrete metrics organized across three dimensions. Each metric is operationally defined with explicit measurement procedures, scoring rubrics, and threshold values derived from DoD acquisition quality standards.

Metric ID	Metric Name	Dimension	Threshold
ACC-01	Semantic accuracy score	Content Accuracy	≥ 90%
ACC-02	Terminology compliance rate	Content Accuracy	≥ 95%
ACC-03	Technical precision score	Content Accuracy	≥ 88%
ACC-04	STE compliance rate	Content Accuracy	≥ 92%
SCH-01	First-pass schema validation	Structural Conformance	≥ 85%
SCH-02	BREX rule compliance	Structural Conformance	≥ 90%
SCH-03	DMC code accuracy	Structural Conformance	100%
SCH-04	Mandatory element coverage	Structural Conformance	100%
SCH-05	Cross-reference integrity	Structural Conformance	≥ 95%
EFF-01	Authoring time (hours/DM)	Operational Efficiency	≤ 6.0 hrs
EFF-02	Review cycles to approval	Operational Efficiency	≤ 2.0 cycles
EFF-03	Rework rate	Operational Efficiency	≤ 15%
EFF-04	Cost per data module	Operational Efficiency	≤ $800

Metric ID	Metric Name	Dimension	Threshold
EFF-05	Throughput (DMs/week/author)	Operational Efficiency	≥ 5.0 DMs

Table 2-2. Evaluation Metrics Framework

2.4 Data Collection Procedures

Data collection was conducted over a 16-week experimental period using a structured protocol. Each authoring session was logged with timestamps, model parameters, prompt versions, and iteration counts. Human expert sessions were recorded using time-tracking software integrated into the Oxygen XML authoring environment.

Content accuracy scoring was performed by a panel of three independent reviewers using a standardized rubric. Reviewers were S1000D-certified technical writers with no involvement in the authoring phase. All data modules were anonymized and randomized prior to review to eliminate order and attribution bias. Inter-rater reliability was assessed at the midpoint and conclusion of the review period using Fleiss' kappa.

Chapter 3 — System Architecture

3.1 Evaluation Pipeline Overview

The evaluation framework is implemented as a Python-based pipeline consisting of four modular components: (1) an LLM Integration Layer that manages API interactions and prompt engineering, (2) an S1000D Validation Engine that performs schema checking and BREX compliance analysis, (3) a Content Analysis Module that scores semantic accuracy and terminology compliance, and (4) a Statistical Analysis Module that computes aggregate metrics and performs hypothesis testing.

The pipeline is designed for reproducibility and extensibility. All configuration parameters — including model selection, temperature settings, prompt templates, and validation rule sets — are externalized in YAML configuration files. The pipeline logs all inputs, outputs, and intermediate results to a structured SQLite database for post-hoc analysis and audit trail compliance.

3.1.1 Technology Stack

Component	Technology	Version	Purpose
Runtime	Python	3.11	Core pipeline execution
LLM APIs	OpenAI / Anthropic / HF	Various	Model inference
XML Processing	lxml / Saxon-HE	4.9 / 12.3	Schema validation
NLP Analysis	spaCy / NLTK	3.7 / 3.8	Content scoring
Statistics	SciPy / pandas	1.11 / 2.1	Hypothesis testing
Data Store	SQLite	3.42	Results persistence
Visualization	Matplotlib / Seaborn	3.8 / 0.13	Reporting

Table 3-1. Evaluation Pipeline Technology Stack

3.2 LLM Integration Layer

The LLM Integration Layer abstracts model-specific API differences behind a unified interface, enabling consistent prompt delivery and response parsing across GPT-4, Claude, and Llama 2. Each model interaction follows a three-phase protocol: (1) system prompt injection with S1000D schema context and BREX rules, (2) task-specific prompt delivery with source material, and (3) structured response extraction with XML parsing and error recovery.

Prompt engineering templates were iteratively refined through a pilot study of 30 data modules (10 per content type). The final prompt architecture uses a chain-of-thought approach that decomposes data module authoring into sequential subtasks: metadata generation, content structure planning, narrative drafting, XML markup application, and self-validation. This approach increased first-pass schema validation rates by 23% compared to single-prompt approaches.

Temperature and sampling parameters were held constant across experimental conditions: temperature = 0.3, top-p = 0.95, and max tokens = 8,192. These parameters were selected based on pilot study results indicating optimal balance between output consistency and content diversity for structured technical writing tasks.

3.3 S1000D Validation Engine

The validation engine performs three levels of compliance checking against each LLM-generated data module:

Level 1 — Schema Validation: Full W3C XML Schema validation against the appropriate S1000D Issue 5.0 schema definition, verifying element structure, attribute values, data types, and cardinality constraints.

Level 2 — BREX Compliance: Business Rules Exchange checking using project-specific BREX data module (DMC-S1000DBRZX-A-D00-00-00-00AA-022A-D), validating allowed values, conditional elements, and notation rules.

Level 3 — Semantic Validation: NLP-based content analysis scoring terminology compliance (STE adherence), cross-reference integrity, and consistency with source technical data packages.

> **NOTE:** The validation engine processes each data module in under 3.2 seconds on average, enabling near-real-time feedback during LLM-assisted authoring workflows.

3.4 Statistical Analysis Module

The Statistical Analysis Module computes all 14 evaluation metrics defined in Section 2.3 and performs inferential statistical testing to assess the significance of observed differences between authoring methods. The module implements the following analytical procedures:

Descriptive statistics: Mean, standard deviation, median, and interquartile range for all continuous metrics, computed per authoring method and content type.

Inferential testing: Paired t-tests for pairwise comparison of each LLM method against human expert baselines. Two-way ANOVA for analysis of interaction effects between authoring method and content type. Bonferroni correction applied for multiple comparisons.

Effect size computation: Cohen's d for all pairwise comparisons, providing a standardized measure of practical significance independent of sample size.

Confidence intervals: 95% confidence intervals computed for all point estimates using bootstrap resampling (10,000 iterations) to ensure robust inference even with non-normal distributions.

Chapter 4 — Results and Analysis

4.1 Content Accuracy Analysis

Content accuracy was evaluated by a panel of three S1000D-certified technical writers using a blind review protocol. Each data module was scored on a 100-point rubric covering semantic correctness (40 points), terminology compliance (30 points), and technical precision (30 points). Inter-rater reliability was assessed using Fleiss' kappa (κ = 0.84, indicating strong agreement).

Authoring Method	Semantic Accuracy	Terminology Compliance	Technical Precision	Composite Score
Human Expert (baseline)	94.2%	96.8%	93.1%	94.7%
GPT-4 Assisted	91.3%	93.4%	88.7%	91.1%
Claude Assisted	90.8%	94.1%	89.2%	91.4%
Llama 2 Assisted	85.6%	87.3%	82.4%	85.1%

Table 4-1. Content Accuracy Scores by Authoring Method (mean %, n=50 per method)

Paired t-tests revealed statistically significant differences between human expert baselines and all LLM-assisted methods ($p < 0.01$ for all comparisons). However, the practical significance was modest: effect sizes ranged from $d = 0.31$ (Claude vs. human) to $d = 0.78$ (Llama 2 vs. human), with GPT-4 and Claude falling within the "small effect" range per Cohen's (1988) conventions. This indicates that while LLM outputs are measurably less accurate than human expert work, the difference is practically manageable through targeted review processes.

Content type analysis revealed that descriptive data modules achieved the highest accuracy across all LLM methods (mean composite = 92.4%), followed by procedural modules (mean composite = 88.7%), with fault isolation modules showing the lowest accuracy (mean composite = 84.1%). The accuracy gap between LLM and human methods widened as content complexity increased, suggesting that fault isolation logic — which requires multi-branch conditional reasoning — remains a challenging task for current LLM architectures.

4.2 Schema Conformance Results

Schema conformance was measured as the percentage of data modules that passed Level 1 (schema) and Level 2 (BREX) validation on the first submission without human correction. Results varied significantly by content type, with descriptive modules achieving the highest pass rates and fault isolation modules the lowest.

Authoring Method	Descriptive	Procedural	Fault Isolation	Overall
Human Expert	98.0%	96.0%	94.0%	96.0%
GPT-4 Assisted	92.0%	88.0%	78.0%	86.0%
Claude Assisted	94.0%	90.0%	80.0%	88.0%
Llama 2 Assisted	82.0%	74.0%	62.0%	72.7%

Table 4-2. First-Pass Schema Validation Rates by Content Type

The most common validation failures across all LLM methods were: (1) incorrect nesting of procedural step elements within `<proceduralStep>` containers (34% of failures), (2) missing mandatory attributes on `<dmIdent>` elements (22% of failures), and (3) invalid enumerated values in BREX-controlled attributes (19% of failures). These failure patterns are addressable through prompt refinement and post-processing rules, as demonstrated in the iterative improvement analysis below.

An iterative correction experiment was conducted in which LLM models were provided with their validation error reports and asked to self-correct. After one correction cycle, overall schema validation rates improved to 93.8% (GPT-4), 95.2% (Claude), and 84.6% (Llama 2), indicating that a single automated feedback loop can substantially close the conformance gap.

4.3 Performance Benchmarks

Metric	Human Expert	GPT-4	Claude	Llama 2
Avg. authoring time (hrs/DM)	12.4	4.8	4.6	5.2
Avg. review cycles	1.8	2.4	2.3	3.1
Rework rate	8.2%	14.6%	13.2%	22.4%
Throughput (DMs/week)	3.2	8.4	8.7	7.6
Avg. cost per DM	$1,240	$520	$485	$580

Table 4-3. Operational Efficiency Metrics Comparison

LLM-assisted methods demonstrated substantial efficiency gains across all operational metrics. Average authoring time was reduced by 58–63% depending on the model, with Claude achieving the highest throughput at 8.7 data modules per week per author (compared to 3.2 for human experts). Cost per data module decreased by 53–61%, driven primarily by reduced labor hours.

The trade-off between efficiency and quality is most apparent in the rework rate metric: LLM-assisted methods required 61–173% more rework than human expert baselines. However, because the initial authoring time is dramatically lower, the net time-to-completion (authoring + rework) still favors LLM-assisted approaches by a factor of 1.8–2.1x.

4.4 Cost-Benefit Analysis

A cost-benefit model was developed for a representative DoD program requiring 1,500 S1000D data modules over a 36-month development period. The model accounts for authoring labor, review labor, LLM API costs, tooling infrastructure, and training investment.

Cost Category	**Traditional Workflow**	**LLM-Assisted Workflow**	**Delta**
Authoring labor (1,500 DMs)	$1,860,000	$742,500	–$1,117,500
Review labor	$465,000	$558,000	+$93,000
LLM API costs	$0	$67,500	+$67,500
Tooling & infrastructure	$120,000	$185,000	+$65,000
Training & onboarding	$45,000	$72,000	+$27,000
Total program cost	**$2,490,000**	**$1,625,000**	**–$865,000**

Table 4-4. Program-Level Cost-Benefit Projection (1,500 Data Modules)

The model projects a net cost reduction of $865,000 (34.7%) for the LLM-assisted workflow, with breakeven occurring at approximately month 8 of the program after initial tooling and training investments are amortized. Sensitivity analysis indicates that cost savings remain positive even under pessimistic assumptions (25% increase in review labor, 40% increase in rework rates).

Chapter 5 — Recommendations

5.1 Implementation Guidelines

Based on the empirical results of this study, the following guidelines are recommended for defense acquisition programs considering LLM-assisted S1000D IETM development:

Phased adoption. Begin with descriptive data modules (highest LLM accuracy) before extending to procedural and fault isolation content types. This approach builds team confidence and allows iterative prompt refinement with lower risk.

Human-in-the-loop review. All LLM-generated data modules must undergo SME technical review. LLM outputs should be treated as first drafts requiring validation, not finished deliverables. The review process should be documented and auditable per MIL-STD-40051-2 quality assurance requirements.

Automated validation gates. Integrate the S1000D validation engine as an automated quality gate in the authoring pipeline. Data modules that fail Level 1 or Level 2 validation should be returned to the LLM for self-correction before human review, as the iterative correction experiment demonstrated significant conformance improvement with minimal additional cost.

Prompt version control. Manage prompt engineering templates under formal configuration management (CM) with revision tracking, change justification, and regression testing. Prompt templates should be treated as controlled project artifacts with the same CM rigor applied to schema customization files.

Data sovereignty compliance. Ensure that all LLM interactions comply with program-specific data handling requirements. For programs handling CUI or export-controlled data, deploy locally-hosted models (e.g., Llama 2) or negotiate appropriate data processing agreements with cloud LLM providers. No classified data should be transmitted to external LLM APIs under any circumstances.

5.2 Risk Mitigation Strategies

Technical accuracy risk. Mitigate through mandatory SME review of all LLM-generated content, with particular attention to procedural steps involving safety-critical operations and fault isolation logic branches. Implement a "confidence scoring" system that flags content sections with low model certainty for priority review.

Schema conformance risk. Mitigate through automated pre-submission validation and iterative LLM self-correction cycles. Maintain a registry of known failure patterns (Section 4.2) and implement targeted post-processing rules to address the most common validation errors programmatically.

Vendor dependency risk. Mitigate through model-agnostic pipeline architecture (as implemented in this study's evaluation framework). The unified API abstraction layer enables rapid model substitution without impacting downstream validation or analysis components.

Workforce transition risk. Mitigate through structured training programs that reposition technical writers as "AI-assisted authoring specialists" rather than replacing them. Emphasize that LLM tools augment human expertise — they do not eliminate the need for domain knowledge, editorial judgment, or quality assurance skills.

5.3 Future Research Directions

This study identifies several high-priority areas for continued investigation:

Extended content type coverage. Evaluate LLM performance on illustrated parts data (IPD), wiring diagrams, and crew/operator data modules, which were excluded from the current study scope.

Fine-tuned domain models. Investigate the performance impact of fine-tuning base LLM models on corpora of validated S1000D data modules, hypothesizing that domain-specific training will improve both accuracy and schema conformance beyond the general-purpose model results reported here.

Multi-modal integration. Explore LLM capabilities for generating or interpreting technical illustrations, schematics, and annotated figures that accompany S1000D textual data modules.

Longitudinal production study. Conduct a 12–18 month field study of LLM-assisted authoring in an active defense acquisition program to validate the cost-benefit projections under real-world operational conditions.

Automated BREX generation. Investigate whether LLMs can assist in the generation and validation of BREX business rule definitions themselves, potentially accelerating the S1000D project setup phase.

References

Anthropic. (2024). Claude Technical Report. Anthropic Research.

ASD. (2022). S1000D International Specification for Technical Publications, Issue 5.0. AeroSpace and Defence Industries Association of Europe.

ASD. (2021). ASD-STE100 Simplified Technical English Specification, Issue 8.

Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences (2nd ed.). Lawrence Erlbaum Associates.

Department of Defense. (2019). MIL-STD-40051-2: Preparing Digital Technical Information for Interactive Electronic Technical Manuals (IETMs), Revision B.

Department of Defense. (2020). MIL-STD-3031: DoD Standard Practice for Technical Data Packages, Revision A.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. MIS Quarterly, 28(1), 75–105.

Meta AI. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

OpenAI. (2023). GPT-4 Technical Report. arXiv:2303.08774.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24(3), 45–77.

RTCA. (2011). DO-178C: Software Considerations in Airborne Systems and Equipment Certification.

Smith, N. B. (2024). Design Science Research Study: Evaluating Large Language Model Applications in DoD/DoW S1000D-Conformant IETM Development [Doctoral dissertation, Colorado Technical University].

Appendix A — Evaluation Metrics Detail

A.1 Content Accuracy Scoring Rubric

Semantic Accuracy (ACC-01) — 40 points

Score Range	Criteria
36–40	Content is factually correct with no technical errors; all statements are verifiable against source TDP
30–35	Minor inaccuracies that do not affect operational meaning; ≤ 2 factual discrepancies per module
20–29	Moderate inaccuracies requiring substantive correction; 3–5 factual discrepancies per module
0–19	Major errors that would compromise safety or operational effectiveness if uncorrected

Terminology Compliance (ACC-02) — 30 points

Score Range	Criteria
27–30	Full STE compliance; all terms from approved SNL; no unapproved abbreviations
22–26	Minor deviations; ≤ 3 non-STE terms; all abbreviations defined
15–21	Moderate deviations; 4–8 non-STE terms; inconsistent abbreviation usage
0–14	Widespread non-compliance; > 8 non-STE terms; undefined abbreviations

Technical Precision (ACC-03) — 30 points

Score Range	Criteria
27–30	All specifications, tolerances, and values accurate; units correct; no ambiguity
22–26	Minor precision issues; ≤ 2 unit or tolerance errors; minimal ambiguity
15–21	Moderate precision issues; 3–5 errors in specifications or values
0–14	Major precision failures; > 5 specification errors; significant ambiguity

A.2 Schema Validation Error Categories

Error Category	Description	Frequency (% of all failures)
Element nesting	Incorrect parent-child relationships in XML structure	34%
Missing attributes	Required attributes absent from mandatory elements	22%
Invalid enumerations	Attribute values not in BREX-approved value lists	19%
Cardinality violations	Incorrect number of child elements (min/max)	11%
Data type mismatches	Element content does not match declared data type	8%
Namespace errors	Incorrect or missing namespace declarations	4%
Other	Miscellaneous validation failures	2%

Table A-1. Distribution of Schema Validation Errors Across All LLM Methods

---

Appendix B — Sample Data Module Outputs

B.1 Descriptive Data Module — LLM-Generated Sample

The following excerpt demonstrates a Claude-generated descriptive data module for a flight management system display unit. The module passed Level 1 and Level 2 validation on first submission.

```
<?xml version="1.0" encoding="UTF-8"?>

<dmodule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"


xsi:noNamespaceSchemaLocation="http://www.s1000d.org/S1000D_5-0/xml_s
chema_flat/descript.xsd">

  <identAndStatusSection>

    <dmAddress>

      <dmIdent>

        <dmCode modelIdentCode="HON72" systemDiffCode="A"

          systemCode="34" subSystemCode="1" subSubSystemCode="0"

          assyCode="00" disassyCode="00" disassyCodeVariant="A"

          infoCode="040" infoCodeVariant="A" itemLocationCode="D"/>

        <language languageIsoCode="en" countryIsoCode="US"/>

        <issueInfo issueNumber="001" inWork="00"/>

      </dmIdent>

      <dmAddressItems>

        <issueDate year="2024" month="03" day="15"/>

        <dmTitle>

          <techName>Display Management Computer</techName>

          <infoName>Description</infoName>

        </dmTitle>

      </dmAddressItems>
```

```
    </dmAddress>
  </identAndStatusSection>
  <content>
    <description>
      <levelledPara>
        <title>General</title>
        <para>The Display Management Computer (DMC) is the primary
          processing unit for the integrated flight deck display
          system. The DMC receives flight data from the Air Data
          Inertial Reference Unit (ADIRU), navigation data from the
          Flight Management Computer (FMC), and engine parameters
          from the Full Authority Digital Engine Control (FADEC)
          system.</para>
      </levelledPara>
    </description>
  </content>
</dmodule>
```

Figure B-1. Excerpt from Claude-generated descriptive data module (DMC-HON72-A-34-10-00-00A-040A-D)

End of Document

> **DISCLAIMER REMINDER:** This document is a hypothetical technical writing sample created for professional portfolio purposes only. It does not contain any proprietary, classified, or confidential information from any organization. All data, findings, and program references are fictional.