Re_forms 21

CHECKLIST:

Identify Your "Hidden Business Rules" Before Migration

Instructions: Use the points below to audit your Oracle Forms environment. If you check **more than 3 boxes**, your project requires an engineering-led **Structural Refactoring** approach rather than just "AI assistance."

1. Language Specificity and Logic (The "PL/SQL Trap")

☐ **Three-Valued Logic:** Does your system rely on distinguishing between **NULL, TRUE,** and **FALSE** states in complex logical conditions? AI often simplifies this to standard binary logic, which can silently destroy data integrity.

☐ **Custom Data Types:** Do you use "exotic" data types or high-precision calculations on **NUMBER** types? AI tends to incorrectly map these to standard Java primitives, leading to subtle but critical rounding errors.

☐ **Complex Triggers:** Does your system utilize cascading triggers where the specific execution order is vital? AI models lack the temporal context to ensure these sequences remain intact.

2. Integration and System Environment

☐ Binary Communications (OLE2 / WebUtil): Does the system communicate directly with Excel, Word, or external hardware via binary protocols? AI cannot "see" these functions in source code because they are stored in compiled **.fmb** binaries.

☐ Cross-Module Dependencies: Does a change in one Forms module affect database procedures in non-obvious ways (e.g., via shared global variables or session state)?

3. "Tribal" Knowledge vs. Documentation

☐ Code as the Only Truth: Is your system documentation outdated or non-existent, leaving the "last Forms developer" as the only one who understands the intent? AI cannot interview stakeholders for business context.

☐ Legacy Engineering Patterns: Was the system built using emerging local practices from 30+ years ago that aren't represented in modern public repositories like GitHub?

4. Operational Semantics

☐ Custom Error Handling: Does the system have its own proprietary exception-trapping logic that determines transaction rollbacks?

☐ Implicit Constraints: Are there critical data validation rules defined deep within the UI layer (Forms) rather than at the database level?

Why This Checklist is Critical

Most AI models are trained on open-source code where business rules are typically separated from the presentation layer. In Oracle Forms, these rules are "entangled"—woven into the UI code and hidden within compiled binaries.

General-purpose AI interprets what it sees, but it cannot interpret what it hasn't been trained to understand. If the AI misses a single implicit constraint or misinterprets a **NULL** logic state, the resulting modern system will be plagued by inconsistent logic that destroys user confidence and operational quality.

The ReForms21 Perspective: No Guessing, Only Evidence

At ReForms21, we don't ask an AI to guess what the original developer intended. We bridge the "Scarcity Gap" by building a comprehensive technical and operational model of your entire system. We iteratively construct a high-confidence map of major business processes based on actual structural evidence, not probabilistic suggestions.

Take the Next Step

Are you ready to perform a professional analysis of your system?

Don't let hidden logic derail your digital transformation. Contact ReForms21 today to receive a tailored Migration Effort Estimate. We will identify your "hidden business rules" with engineering precision and provide you with a roadmap built on certainty, not hallucination.

Book your demo with us: click here.

re_forms 21