CLIENT	**QRA Corp**
PROJECT	**Implementation guide**
OBJECTIVE	**Increase inbound traffic to the client’s website and compel readers to try out client’s tool to automate their own processes.**

COPY EXCERPT

Automating the INCOSE Guide for Writing Requirements

QRA

Automating the INCOSE Guide for Writing Requirements

A time-saving guide for system engineers and requirements analysts

Call or write CopyEngineer for a PDF of the complete guide.

Or view it online at: qracorp.com/incose-requirements-guide/

Writing Clear Requirements Isn’t Easy

Most companies understand the need for specifying clear, unambiguous requirements, both for the systems they build and sell, and for the systems they commission and procure from other vendors. Numerous studies have shown that the cost of fixing engineering errors in systems and software increases exponentially over the project life cycle (Jonette[i], Boehm[ii], Rothman[iii], McGibbon[iv], Chigital[v]), and that more than half of all engineering errors originate in the requirements (Martin[vi]).

Writing clear, unambiguous requirements in natural (spoken) language on a consistent basis, however, is not easy.

“Natural language’s extensive vocabulary and commonly understood syntax facilitate communication and make it an inviting choice to express requirements,” says William Wilson, a former principal systems engineering consultant with the Software Assurance Technology Center (SATC). “The informality of the language also makes it relatively easy to specify high-level general requirements when precise details are not yet known.

“However, because of differences among formal, colloquial, and popular definitions of words and phrases and the effort required to produce detailed information, these same attributes also contribute to documentation problems. The use of natural language to prescribe complex, dynamic systems has at least three common and severe problems: ambiguity, inaccuracy, and inconsistency.” [vii]

The Drawbacks of the Best Checklist for Requirements Review

These problems have long been recognized, and systems engineers and have sought solutions to them for years. Until recently, most of these solutions have fallen into two classes: (1) the use of abstract notations to eliminate natural language from specifications, and (2) the use of guides, rule sets and checklists for writing better natural language requirements.

Unfortunately, each of these solution classes comes with problems of its own.

In this guide, as our title suggests, we will focus on overcoming the drawbacks of what is perhaps the most well-known, respected, and widely used third-party requirements quality guide: the *Guide for Writing Requirements* published by the International Council on Systems Engineering (INCOSE).

> *“It’s rare that an outside writer can also help with subject matter expertise. John contributed not just great writing but valuable insight, as well.”*
>
> Trevor Bradley
> Marketing Manager
> QRA Corp