

Chapter 1

Automation Projects and Legal Risk

At first glance, the legal risks facing the automation industry would appear to defy categorization. Are such projects about instrumentation? Are they about services? The answers to these questions are varied. One common way to speak of automation projects is to say that they involve “delivery” of equipment; another is to characterize them as the “sale” of a product. Yet another is to speak in terms of the “installation” of industrial computers or the “development” of software. While all these classifications are more or less accurate (not to mention applied with some frequency), none of these fully captures the essence of this type of endeavor—and therefore none points to the optimum means of managing legal risk.

The best way to approach risk management in automation is to treat such projects as a very specialized type of *construction project*. Not only does this approach permit examining these projects in a somewhat more dynamic and sequential (as opposed to static and scattered) way, it makes it possible to apply a very useful and well-established vocabulary—even if it is a vocabulary more traditionally reserved for projects involving pouring concrete and erecting curtain wall.

Yet there may be something of an industry bias against this approach. Even in the most greenfield of projects, it is not unusual to see a bright line imposed between the construction side of the project and the MRO (maintenance, repair and operations) side—often including parallel lines of authority and completely different personnel. Leaving aside for a moment the incongruity of the “M” and “R” parts of the MRO acronym in the greenfield context (i.e., because the maintenance and repair activities in a greenfield project would seem to be few in number), the split does provoke a question that is worth asking: are the risks that much different? The answer is “No.” The risks in both are imposed by contract.

Speaking the Language of Contract

Realizing that automation projects must navigate the world of construction contracts does not exactly require a leap of faith. So why do automation practitioners frequently deny this fact? One reason is the lingo. “We don’t have a contract—we just have this purchase order,” is what more than one control system integrator has told me. Yet as any first-year law student quickly learns, an exchange of proposal and purchase order is not the opposite of a contract—it is just another *type* of contract. In other words, it is a member of the same family as a thirty-page, single-spaced document with formal signature blocks on the last page.

But here is perhaps the most difficult part to understand: very few automation companies recognize the fact that the legal expense of unraveling a dispute involving the type of contract reflected in a simple purchase order is often much greater than for a contract with 30 pages of lawyer-speak. Why is such a dispute more expensive? The reason is so obvious that it tends to be completely overlooked. In the proposal and purchase order realm, the legal relationships, obligations and rights of both parties are much less clearly defined. So when a dispute erupts, you will be paying your lawyer to assert what you *meant* to say instead of what you *did* say. That takes more time—and time, of course, is money.

The good news is that legal risk in automation projects can be successfully managed by using the language of contracts at the front end. Now for the bad news: there are two distinct vocabularies (let’s call them “dialects”) in this area that must be mastered to maximize project success.

The Dialect of the Traditional Construction Contract

The first dialect to be mastered is that of the traditional construction contract. This dialect—which speaks of change orders and indemnity and “liquidated damages”—is important for two reasons. First, it is the dialect nearly everyone else on the project will be speaking (and therefore it is an advisable way of communicating with those persons). Second, it is a very old dialect, with a great many useful terms with application to automation projects. Among the traditional construction contract terms of maximum importance (each of which will be explored in greater detail in the coming chapters of this book) are the following:

- **Integration, Incorporation by Reference, Order of Priority and Dragnet Clauses.** These terms refer to one of the most fundamental—and (ironically) neglected—questions of all. What scope of work was agreed upon? What happens when one part of the contract disagrees with another? What happens when agreement on a particular point is unclear?
- **Change Order Clauses.** What are the options when scope is arguably added to or subtracted from the project? Can these changes be unilaterally imposed?
- **Payment and Pay-When-Paid Clauses.** How does payment work? What conditions must be met for payment to be made? Does it matter if others are not paid?
- **Liquidated Damages, Force Majeure and No-Damages for Delay Clauses.** What are the risks and options when the project takes longer than expected? Does it matter whether the delay was extraordinary or routine? What if there is a suspension of all activity? Does it matter whether it was an “act of God” or that someone was responsible?
- **Limitation of Liability and Consequential Damages Clauses.** If something on the project goes wrong, are there limits to the potential legal liabilities? Could one of the participants held responsible for the lost profits of others?
- **Indemnity Clauses.** Did any of the participants agree to extend a “protective shield” over the project in any respect? What problems does this shield protect against?
- **Warranties.** What is the duration and scope of any warranty? Are warranties implied by the law but not stated in any written part of the agreement? (Yes, you heard me correctly—agreements implied by circumstances, but not written down.)
- **Claim, Mechanic’s Lien, Bond, Retainage and Dispute Resolution Clauses.** How are disputes to be resolved? Are liens an option or is the project bonded (or both)? How are retainage amounts (withheld funds) released—and can claims be made upon the retainage of another? How and where must disputes be resolved?

Knowledge of all these terms—and the contract paragraphs in which they are defined—goes a long way toward successfully managing risk in an automation project. Still, if automation risk management were a course of study, mastering these terms would yield only a middling grade, because you must also

know the language of risk that is particular to these projects and few others—namely, the dialect of automation.

The Dialect of the Automation Contract

The exercise of automation law requires new words and new approaches (beyond those supplied in the traditional construction project) in at least three areas. Those areas are changes, performance and intellectual property.

Changes. It has been said that construction contracts are more challenging than other types of contracts because, unlike the other contracts, construction contracts are being written at the same time as they are being performed. While this is a bit of an exaggeration, the underlying point is worth bearing in mind. It is simply an acknowledgement that modifications of contract terms—most often in the form of agreed-upon changes or as a result of unforeseen events—are not the exception to the rule, but are the rule. Indeed, it would be a most unusual construction project (perhaps there are none) in which the original specifications, terms and conditions remained unchanged through final completion.

Take that reality, multiply it times ten, and you will have a sense for the multiplicity of changes that automation-related companies (whether end user, system integrator or engineering firm) typically confront. By their very nature, automation projects are constantly changing. For maximum success in managing automation project risk, therefore, you must find a way to manage this anticipated, but frustrating state of affairs.

The flood of changes can be managed in at least two ways. The first way is to control the definition of “change.” Because the final scope of the software component, for instance, of an automation project is an inherently mobile target, it is in the interest of owners and end users to limit the definition of “change” to a certain threshold or criteria. Meanwhile, it is in the interest of control system integrators and engineering firms to avoid any such limitation, while carving out a point at which excessive changes are deemed a breach of the contract (in traditional construction parlance, this is known as a “cardinal change”).

The second way is to control how adjustments due to change are *processed*. As in the first case, owners and end users will want to impose strict notice criteria and tight timeframes for seeking adjustments. Integrators and engineering firms, on the other hand, tend to benefit from relaxed or nonexistent notice provisions and deadlines.

Because the occurrence of changes is so predictable on automation projects, the importance of addressing changes at the front end cannot be overemphasized.

Performance. Automation contracts are also different from traditional construction projects because they almost always involve the construction of a system with moving parts. For that reason, specifications often do not measure performance by static criteria (e.g., was this thing built in conformity with these particular dimensions and consisting of these specified materials?) Instead, the specifications have a tendency to be either output-based (does the thing that was built produce 100 widgets per hour?) or they are satisfaction-based (is this thing that was built fit for its intended purpose?) The first type of specification is called a “design spec,” and the second two are variations on what is called a “performance spec.”

The tug-of-war between owners and automation contractors is also (predictably) present here. Naturally, owners want to be satisfied, and should push for the most subjective performance spec that is possible. Automation contractors, on the other hand—especially if they are fulfilling a design that is largely dictated by an owner’s consultant—will want to limit their construction commitment to building that which was specified without any guarantee as to performance. Prevailing in this struggle is not easy. It can literally mean the difference between a project that is labeled by the contract terms as a success or a failure.

Intellectual Property. Because automation is a subspecies in the proliferating realm of computer technology, the automation industry is naturally evolving at a faster rate than the traditional world of “sticks and bricks.” New ways of solving problems do not appear in fits and starts, but in nearly every project. Indeed, one way of thinking about the automation industry is as a type of engine for solving problems—one that routinely asks, for example, “How might we best control this process?” The result is often not just an elegant solution, but a valuable one.

But who owns the solution? Is it a “work for hire,” paid for by the owner or end user and therefore entirely within its ownership and control? Or is it a valuable invention of the system integrator that must be retained because it is vital to its future success as a business? What happens to the underlying legacy intellectual property of an integrator that may be incorporated into the design? What about commercial off-the-shelf software that is included? Are there limitations to the future uses of the application—or have the parties decided to license the technology in some way between them?

Obviously, the answers to these questions are enormously important, not just for project success, but for long-term *enterprise* success on the part of both system integrator and owner.

Knowing the Neighborhood

While it may be all well and good to know (and to speak) both the traditional construction contracting dialect and the more specialized automation one, such fluency only goes so far when you are operating in an atmosphere in which people would prefer not to engage you in a conversation.

You: “Let’s talk about terms.”

Them: “Sorry, we are not permitted to change these terms.”

Welcome to the world of the take-it-or-leave-it contract.

Both owners and automation companies may have good reasons for this attitude. First, it is, without question, easier. Saying “we are not permitted to change these terms” beats the back-and-forth of negotiations (and, heaven forbid, dealing with lawyers) every time. Second, market leverage or business conditions (“you need us more than we need you”) may make taking such a position all too tempting—the contractual equivalent of “shooting fish in a barrel.” Third, it minimizes risk for the companies that successfully impose their own terms.

Naturally, there are tactics for dealing with this approach if your company happens to be on the receiving end. Among them: bidding with exceptions, providing balanced (i.e., non-threatening) alternate terms, appealing on the basis

of fairness, making the argument that risk should reside with the company with the most control over that risk, or even—when all else fails—incorporation of a proposal, rider or other document into the larger contract that may round off the sharpest edges. (We will explore each of those tactics in more detail later in this book.)

But even if there is no negotiating—not even a conversation—that does not mean that knowing the language of contracts is without value. There are always two decisions that, regardless of negotiating stance, cannot be taken away from any party in an automation project: the decision on price and the decision as to whether to participate in the project in the first place.

That is why it is a central argument of this book that the contract dialects must be learned—and that attention to these dialects must be paid. Whether it involves establishing a process for non-lawyer company managers to analyze the “paper” of a deal or involves hiring lawyers to do so is a matter of ability and preference. The important thing is that someone performs a focused review.