Preventing and Resolving Software Implementation Disputes: Lessons for Contracting

From the Experts

Robert Kriss and Brad Peterson

This is the last in a series of three articles that address how to prevent and resolve software implementation disputes. The first article considered proactive measures that could get the project back on track without litigation. The second article discussed aspects of litigating a dispute. This article will provide ideas on how to use contract provisions to reduce the risk of disputes and increase the likelihood of a fair result in litigation.

Our hypothetical transaction involves a medical center (Good-Health) licensing SoftwareCo's software (ChartX) to input and retrieve clinical and cost data for patients. As described in the previous articles, the software performed too slowly to be usable and produced inaccurate cost data. SoftwareCo blamed GoodHealth for failing to provide accurate information to assist in the integration of the systems. SoftwareCo also argued that it never agreed to the performance criteria that GoodHealth proposed after the contracts were executed and that performance was slowed by required customization. As to damages, SoftwareCo contended that its implementation work

istockphoto.com

met industry standards so even if the standard software was somehow defective, the only damage recoverable was the license fee, which was a fraction of what GoodHealth paid SoftwareCo and others to implement ChartX.

What did GoodHealth do differently in contracting with a replacement vendor?

First, GoodHealth included functional, performance and other acceptance criteria in the warranty provisions of both the license and implementation agreements, each clear enough to test. Though terms such as "reasonable," "customary" or "industry standard" were seductive short-cuts in the first contracts, GoodHealth delayed the new contracts until the criteria could be pinned down. Particularly with respect to performance, where factors unique to GoodHealth affected speed of operation, those terms provided little or no guidance to

the parties during settlement negotiations or to the court in litigation, increasing the risk of disputes and an unfair result in litigation.

In addition, GoodHealth's new contract made it clear that both the standard software and the software as implemented must meet specified acceptance criteria. When deciding whether to license the new software, GoodHealth asked to see results of the vendor's internal testing to assure itself that the standard software worked as represented before customization. In addition, GoodHealth secured an initial 60-day "acceptance period" for the standard software with a right to exit without cost if the software did not meet the pre-customization acceptance requirements.

GoodHealth made the implementation agreement clear that the vendor accepts responsibility for meeting the acceptance criteria, notwithstanding the customization, at the expected cost or informing the customer when new requirements cannot be implemented within the acceptance criteria or expected cost, at which point customer can exit or amend the agreement. GoodHealth decided to invest in rigorous contract governance for the implementation to be sure that these protections were actually used and delivered the anticipated value.

The contracts also made clear that meeting the acceptance criteria is within a fixed or capped part of the price. A promise to meet acceptance criteria at an unlimited time-and-materials cost is no promise at all. With a pricing approach that shares the risk of unknown implementation requirements, the new contracts provide incentives to control cost, timing and performance.

Second, GoodHealth recognized that implementation would be a collaborative effort to build a system that meets the acceptance criteria. But, GoodHealth is relying upon the vendor's expertise to build the system correctly. Disputes often arose in the first contract when the parties blamed each other for shortfalls in the system and cost overruns, often after substantial cost overruns and delays occurred.

To avoid good faith misunderstandings and to prevent a party from wrongfully blaming its failures on the other party, GoodHealth included a "savings clause" in the second contract. That "savings clause" requires the vendor to provide written notice to a specified GoodHealth representative when the vendor believes that GoodHealth is acting or failing to act in a manner that both violates GoodHealth's obligations under the agreement and may prevent the vendor from meeting its obligations, to give GoodHealth a reasonable opportunity to correct the problem following notice, and, in any event, to try to perform despite the problem. The clause saves GoodHealth from being in breach if the vendor fails to do so.

The "savings clause" gives GoodHealth's management an opportunity to direct its personnel to improve their performance and thus avoid unnecessary and/or incorrect work by the vendor. Alternatively, GoodHealth may disagree with the vendor's assessment and that disagreement can be resolved before unnecessary or incorrect work is performed. Finally, the savings clause gives the vendor a strong incentive to identify and flag problems when they happen and reduces the risk

that the vendor will claim a GoodHealth failure as an excuse after it is too late to even tell whether that failure caused a problem.

Finally, GoodHealth included a provision stating that if the vendor fails to deliver software that meets acceptance criteria within a specified period of time, the customer has the option of terminating the agreement and obtaining a refund of all amounts paid to vendor under both contracts and any other costs incurred in developing or configuring other systems in reliance on the software meeting the warranty specifications. This is intended to address the risk that damages are barred as consequential or subject to a low damage cap in the license agreement.

These are just a few examples of the numerous upgrades that GoodHeath made to its IT contracting based on what it learned from informal and formal dispute resolution. By using what it learned from both contract and litigation lawyers, GoodHealth put itself on the path to a smooth, successful, well-run software implementation.

Robert Kriss and *Brad Peterson* *are partners at Mayer Brown. Based in Chicago, they graduated from Harvard Law School and they have worked together on technology deals and disputes for over 20 years. Kriss is a seasoned trial lawyer specializing in technology disputes and Peterson helps clients with software, digital, outsourcing and data deals.*