

5 Signs Of Trouble For Customers In Software Contracts

By **Robert Kriss and Brad Peterson** (April 25, 2018, 11:50 AM EDT)

Implementing new software is a challenging process, even with the best vendors and contracts. In the course of assisting clients with getting vendor relationships back on track and in representing clients in litigation when the relationship has ended unhappily, we have identified a number of potential early warning signs of trouble for customers in software contracts. In our experience, customers can mitigate the risks if they monitor these warning signs from the beginning of the implementation through going live.

1. Dissent or Lack of Cooperation Among Stakeholders who were not Involved in Negotiating the Contract

Sometimes the people who will be using the new software did not have much input in developing the contract requirements. After the project is underway, they begin to voice objections or even attempt to undermine the project because they do not like the planned functionality or the changes that the new software will require them to make.

Mitigation

In most cases, it will be better to seek input from all major stakeholders before the contract is executed. However, if that was not done, then change management activities should be initiated early before time and money are wasted. If there are serious objections from stakeholders, the objections likely will surface during change management. If users have valid objections or ideas for improvement, the contract should be renegotiated, if necessary, or if the contract is flexible enough, then new requirements should be given to the vendor. The key to success is to keep the written contract aligned with the customer's needs so that if what is delivered does not meet those needs, the customer can pursue correction or some other remedy under the contract. Follow the formalities required to amend the contract as specified in the contract or, when the contract is silent, applicable contract law. For example, a flurry of emails with attached draft requirements and nothing physically or electronically signed at the end of the string may not create any legally enforceable obligations.

2. Frequent Changes in Business Requirements Provided to the Vendor



Robert Kriss



Brad Peterson

In some cases, frequent changes in requirements may be sensible because they reflect material new information (such as the outcome of the prior sprint in an agile project). However, frequent changes in business requirements may sometimes be the result of rushing to a solution without a consensus on what problem the software is to solve. Also, customization of standard software and late changes in new software can adversely affect the cost and performance of the software.

Mitigation

Discuss with the vendor the volume and type of changes being made to determine whether the vendor believes the changes are creating problems with respect to cost estimates, milestone dates or performance of systems. Document the discussions in communications with the vendor contemporaneously so that if cost overruns, delays or performance problems arise later that the vendor did not warn about earlier, the vendor cannot blame the problems on the customer. In addition, in some cases it may be better to lock down requirements and determine whether the system can function and perform adequately without additional customization. In any event, the key to success is to create an accurate written record showing who is responsible under the contract for any additional charges or delay caused by changes in business requirements.

3. Lack of Clarity as to the Identity of the Vendor's Team or Vendor's Delay or Refusal to Provide Organization Charts or Background Information on Qualifications of Key Personnel such as Solution Architects.

Sometimes after winning a contract by offering a low price, a vendor attempts to cut corners to save cost by assigning too few or poorly qualified personnel to complete the project and that results in unsatisfactory software and a contract dispute over who is responsible for the costs of repair..

Mitigation

Request current organization charts and notices of changes in personnel even if the contract does not expressly provide for that right. For certain positions, request background information. Solution architects may, for example, be particularly important to project success because they are responsible for laying out the basic configuration of the system and making sure that all components will work well together and not create performance bottlenecks. Also, if standard software is being used as the core of the system, it may be important that key personnel have experience with that standard software. If the people are not adequate to the task, send firm correspondence under the contract demanding an adequate team.

4. Documentation Problems, such as Draft, Unsigned Business Requirements Documents, Absence of Clear Software Design Documents, Absence of Detailed Logical and Physical Data Model Schemas, or Delayed Testing Plans

Effective software design is not done solely in a coder's head. Whether a waterfall or agile approach is used, clear documents will help the implementers and coders to configure standard software and develop the custom software that is necessary to meet business needs. When these documents are not prepared, or the vendor is delaying in providing copies for customer review and comment, that may mean the process is not being conducted in an effective manner and delays, cost overruns and performance problems may arise.

Mitigation

Have a software development expert on staff or retain a consultant to review documentation and process as the project proceeds to, among other things, verify that the work continues to match the contract. With this limited scope, the cost of such assistance should be modest compared to the benefits.

5. Lack of Agreed Performance Criteria Before Work Begins

We have seen instances in which the original contract does not contain performance criteria (speed and volume throughput of operations). Some contracts provide that user acceptance test criteria will be developed at a later stage in the process. If the customer and supplier discover that the system is performing slowly before the parties reach formal agreement on testing criteria, the customer and supplier may never reach agreement on the criteria, or the supplier may demand additional time and money to meet the “new” criteria. If the system shortfalls are discovered after the software is coded and testing criteria are agreed, and if contract pricing is on a time and material basis, the customer could bear the substantial cost of rework and delay.

Mitigation

Agree on performance criteria as soon as possible with clear quantitative units and related system requirements. Even if the agile methodology is followed and changes are made to functionality during the course of a series of sprints, it is important for the vendor to be able to evaluate whether the requested changes in functionality will adversely affect performance or require more costly hardware to meet the performance criteria. If trade-offs between functionality, performance and hardware cost and capacity must be made, it is better to agree on them before the work progresses, rather than through a contract dispute after performance problems appear during testing. At that point, the optimal solutions may no longer be available and the only options may be to engage in costly rework or to acquire more expensive hardware.

Conclusion

If any of these early warning signs appear, it would be prudent for the customer to involve legal counsel to determine the most effective way to orchestrate and document the response. Conduct and statements made by the customer team can be relevant in determining the original intent of the parties and the meaning of contract terms if litigation ensues. Therefore, it is important that members of the customer team build a consistent written record that supports the customer’s position and that team members avoid making statements that the vendor could use to support its position. Also, well-framed written requests for information from the customer coupled with vendor responses can be used as evidence to establish important facts in litigation. A well-structured exchange can produce a couple of key documents that tell the story rather than having to piece together the facts by reviewing hundreds or thousands of emails from dozens of people after the fact. Involving legal counsel thus helps to deliver the intended benefits of the contract, including having the leverage and options that a customer may require if the software project becomes troubled.

Robert J. Kriss and Brad L. Peterson are partners with Mayer Brown LLP in Chicago.

The opinions expressed are those of the author(s) and do not necessarily reflect the views of the firm or Portfolio Media Inc., or any of its or their respective affiliates. This article is for general information purposes and is not intended to be and should not be taken as legal advice.