

THE BUG LIST: A CHECKLIST OF SYSTEM-CRASHING FLAWS IN SOFTWARE DEVELOPMENT AND LICENSE CONTRACTS

A Guide Brought to you by WOOD HERRON & EVANS LLP

You've already done a lot of work: creating your requirements definition, identifying and qualifying prospective vendors, evaluating their proposals, selecting a winner and doing your due diligence, negotiating refinements to their proposal. But before you sign the vendor's form agreement, be sure and measure it against this list of common shortcomings that can, if not corrected, leave your expectations unfulfilled.

What are you really getting?

Contemporary software applications are complex, and ever evolving. You want to be sure that the license accurately specifies:

- Which applications you are getting (including which versions or releases, and whether subsequent versions and releases are included in the support package);
- Whether it will run in your existing environment (hardware, operating systems, network systems, interfaces, operating environment, and infrastructure) and if not, who pays for required improvements;
- Whether you are getting the software in executable form only (required to run the hardware, but indecipherable to humans) or whether you are also getting source code (essential for doing your own customization and maintenance);
- What sort of documentation will be provided (user manuals, system administrator manuals, maintenance manuals, etc.).

It's undoubtedly correct to say that if you don't get this right, nothing else will matter. But as a practical matter, it is unlikely that you as a business person will be able to provide the details. Much of this will take the form of schedules and exhibits to the agreement (requirements document, RFP and response, published documentation, proposal, etc.). Your task is to ask the right questions of the right people to be certain that both you and the vendor have a common understanding and that the details have been adequately memorialized in the

agreement.

And as complex as all of this sounds, understanding the scope of your ownership rights or license can be even more bewildering:

- If it's a custom application, will you own it (or will it be available to your competitors)?
- Is your license limited to a particular machine? a particular size of machine? at a particular location? can you outsource the operation?
- Is it limited to your internal business purposes only? can you service customers or business partners?
- Is it limited to a certain number of users? are these concurrent users? is it a site license? an enterprise license?
- Is your license transferable or assignable? can you transfer it if you sell your business or a business unit? Is there a transfer fee if you do?

Using your software in ways not permitted by your license will almost certainly constitute a material breach of your license agreement (probably voiding your warranties and in some cases triggering an automatic termination of your license). Moreover, such activity may well constitute an infringement of the vendor's copyrights in the software.

What will it really do?

If you've made your purchase decision based on the performance representations made in the vendor's sales literature or made by the vendor's sales agents, you may be in for a disappointment. Make sure that you understand exactly what the license agreement says about what the software will do and that, if the license references user documentation in this regard, that you look to see what that documentation says about functions, features, and performance (including meaningful, measurable response time parameters) and do not simply rely on what you have been told. If you have spent the time and effort to work up an RFP and negotiate

a proposal in response, **be sure** that the contract includes an **affirmative obligation** on the part of the vendor to deliver a system that complies with that documentation.

What will it really cost?

Is the license offered for a one-time fee or must it be periodically renewed? In either case, the license fee may be only a part of the cost of acquiring a new application.

- Will some customization be required in order to tune the software to your business.
- Will there be separate installation and implementation fees?
- Will your staff have to be trained?
- Can you make your own copies of training and user documentation or do you have to buy copies from the vendor?
- Will existing databases have to be converted?
- Will new interfaces need to be written?
- Are all of these services included in the fees you have been quoted? If so, are these fees based on estimates or are they fixed, or not-to-exceed, costs?
- Are there separate support and maintenance costs?
- Does the maintenance agreement have an escalator provision?
- Will you get advance notice of cost increases?
- Will you have to purchase upgrades periodically in order to continue your maintenance and support service?
- Is the system scalable? If you have to go to a larger server to support growth in your business, will you have to buy a more expensive license? If so, can you get a guaranteed upgrade fee now?

Don't overlook these additional costs as they may outstrip the initial license fees.

What happens if you have a problem?

Try as you might to diligently check out the vendor and the product, and as careful as you might be in reviewing the license agreement, contemporary software applications are complex and error free operation should **never** be assumed.

- Make sure that payments are directly tied to deliverables or progress against objectively measurable milestones and that you do not get the payments out in front of the progress.
- Review the software agreement with the conviction that the installation and implementation will not go exactly as planned and hold back a sufficient portion of the fees to provide meaningful financial leverage to ensure the vendor's attention to your concerns and full cooperation in getting them resolved.
- Does the vendor have a finite amount of time to repair

identified defects or deficiencies?

- Is there a drop-dead date for go-live? Have you left enough cushion in your implementation schedule to deal with problems?
- Document every missed milestone or failed deliverable and be careful that you do not inadvertently extend performance deadlines by establishing a pattern of acquiescence.
- Don't fall into the trap of assuming that the first delay or failure is an aberration and failing to object in writing in the interest of preserving your relationship with the vendor. If it turns out not to be an aberration (but instead a harbinger) you may have forever missed an opportunity to start the clock on any cure period.
- Make sure your objections are sent in a manner that is calculated to both notify the owner of the problem and satisfy the specific notice requirements in your contract.

Assume that, having completed your implementation and having switched over to your new system, you will experience an error, failure, or disruption at the most inopportune moment -- satisfy yourself that, under those circumstances, the remedies laid out in the agreement would be sufficient to allow you to sleep at night and that's just what you will be able to do.

- Is there a monetary remedy in the event the vendor can't make it work?
- Do you have to surrender the system in order to get access to the monetary remedy? Are you allowed to retain the system for long enough to make an orderly transition to a substitute?
- Does the contract include a limitation on the vendor's liability -- to a refund of license fees only (common)? or do you get back all sums paid? What about reimbursement for data and file conversion costs?
- Is a copy of the source code in escrow? Will it be current? Will it be useable by you?
- If failure would be catastrophic, do you need to consider insuring against this risk?



For more information, contact:

Steve Gillen

Wood Herron & Evans LLP
2700 Carew Tower
441 Vine Street
Cincinnati, Ohio 45202-2917

sgillen@whe-law.com
Direct Dial (513) 707-0470

Copyright © 2001, 2016 by Stephen E. Gillen
All rights reserved.