Exhibit B: Assessing the Need for Software Maintenance Contracts

Whether the software is new to the agency, or has been in use at the agency for a period of time, agency managers should assess the need for maintenance contracts by answer the following:

- What software maintenance activities are necessary? What software maintenance activities does the business unit want that may not be necessary? Is the agency willing to pay for wants in addition to needs?
- Is vendor-provided maintenance required to ensure that software is continually updated with necessary updates and patches, including security?
- Does the agency have personnel on staff with the knowledge, skills and abilities to conduct software maintenance?
    - If not, can the agency
        - Train its employees for those required knowledge, skills and abilities?
        - Hire employees with the required knowledge, skills and abilities?
        - Tap into existing peer-to-peer support groups? What is the availability and accessibility of suitable user communities?
- Does the vendor plan to release future upgrades to the software that would be provided under a maintenance agreement? Will the agency benefit from the anticipated enhancements to this software? Can the agency reasonably anticipate how often the vendor will release upgrades? Does the agency have the resources to be able to install the future upgrades when they become available?
- How often will the agency use the software? How critical is it to the agency's operations? How long can the agency be exposed to unaccepted consequences of the risk of data loss, system security breaches, erroneous data outputs or system failure?
- How many end users will there be for the software? Will this amount change? What impact, if any, will this have on maintenance needs?
- What is the appropriate response time to address software problems without facing unacceptable consequences (e.g., two hours, four hours, next business day, other)? What is the expected life of the software and the cost of maintenance over that life? How does that compare to the cost of obtaining new software and maintenance?

When considering going without software maintenance, first:

- Review system dependencies and identify all elements of infrastructure involved.
- Review incident history.

- Identify what access the organization has to diagnostic tools, scripts and source code.
- Ensure archive procedures are working to mitigate the effects of excessive data.
- Test live backup and restore scenarios thoroughly.
- Revise event management/monitoring rules to identify out-of-tolerance performance early.
- Define and schedule all routine activities (re-indexing, archiving, metadata refreshes, etc.).

Then consider the feasibility of:

- Isolating applications from the hardware stack through virtualization;
- Rolling back operating system, database, application server, etc. to previously stable versions.
- Establishing a test environment to test changes before being applied to production.
- Segregating the application stack from the environment.
- Implementing a secure ring-of-steel to manage risk as the operating system and sub-systems become vulnerable.
- Fronting the system with trusted domain-based proxies to restrict access, etc.
- Revoking system administration access from all but the few needed to manage master data.
- Implementing a change freeze - All changes must be approved by senior management.
- Informing users that the application is going to remain unchanged going forward and manage the users' expectations.
- Treating all configuration changes (including seemingly trivial routine metadata updates) as formal changes using the change management process.
- Designing a method to identify and protect the agency against unauthorized changes to the software (e.g., use low-level forensic discovery to audit system settings to the INI file level and treat every unplanned change as a threat).

Return to G-Bulletin