Best Practices for Information Systems Software Acquisition and Implementation

June 2003

Office of the City Auditor
Portland, Oregon

CITY OF
PORTLAND, OREGON
OFFICE OF THE CITY AUDITOR
Audit Services Division

GARY BLACKMER, CITY AUDITOR
Richard Tracy, Director of Audits
1221 S.W. Fourth Ave., Room 310
Portland, OR 97204

(503) 823-4005 FAX: (503) 823-4459
www.portlandonline.com/auditor

June 10, 2003

TO: Mayor Vera Katz
Commissioner Jim Francesconi
Commissioner Randy Leonard
Commissioner Dan Saltzman
Commissioner Erik Sten

RE: Best Practices for Information Systems Software Acquisition

Attached is Audit Services Report #298 on best practices for software acquisition and implementation. We believe that this report will be a useful guide for City managers who are responsible for acquiring and developing information systems.

After completing our research, we identified some areas that City Council may wish to address in order to improve the management of software acquisition and implementation. With assistance from the Bureau of Technology Services, which has responded to these suggestions in greater detail, the Council should consider:

1) Adopting administrative rules for software acquisition to ensure that proposals receive thorough consideration by individuals with business, financial, risk management, legal, technological, and operational expertise.

2) Requiring a complete application review before authorizing or executing contracts to acquire or develop the software.

3) Developing increased capacity in the City to perform legal and technical reviews of information systems contracts.

We appreciate the cooperation and expertise received from the Bureau of Technology Services, the Office of Transportation, the Bureau of Development Services and others with whom we worked in conducting this study.

Gary Blackmer
Portland City Auditor

Audit Team: Richard Tracy
Sharon Meross

Best Practices for Information Systems Software Acquisition and Implementation

June 2003

A report by the Audit Services Division
Report #298

Office of the City Auditor
Portland, Oregon

Table of Contents

Introduction

Background

The City of Portland depends upon many ***information systems*** to provide services to the public. "Information systems" (IS) include software applications and hardware that support critical business and administrative operations including emergency communications, tax and revenue collection, and payroll and accounting activities. Most City services, from traffic light control through park recreation services, are supported by various information systems.

To ensure that these critical business functions are reliable and well run, the City must acquire, implement and maintain its informations systems effectively. The risks of failure, lost time and increased public expense require that organizations develop sound guidelines and policies for software acquisition, development and implementation.

The City of Portland's Bureau of Technology Services (BTS) has recognized the need for standardized policies and guidelines for both purchasing and internally developing software. BTS has developed an Application Review/Evaluation Process for proposed information technology projects, and is working on a Systems Development Methodology for both licensed and internally developed applications. One goal of BTS's recent reorganization is to build support and understanding of the importance of IS planning, policies, and sound project management techniques. BTS hopes to promote and incorporate these processes within City bureaus via its Bureau Business Representatives. This report complements BTS's efforts to improve IS management by

providing top management with a nontechnical guide to acquiring and implementing new information systems.

In addition, the City experienced significant difficulties implementing the water and sewer customer information system purchased by the Water Bureau in 1998. Council expressed a desire to understand how problems occurred and ways to avoid future IS problems. By documenting best practices for the acquisition, development and implementation of information systems, this report addresses Council's information needs.

Study objectives, scope, and methodology

The objective of this study was to create a guide that can be used by City Council, bureau directors and other management staff when overseeing a major information systems software project. To create this guide, we reviewed literature pertaining to best practices on the acquisition, development and implementation of information systems software. This literature included publications by the Office of Management and Budget, the U.S. Department of Defense, the American Management Association, various university programs involved in information management issues, and private information technology consultants and consulting groups.

We interviewed two senior managers in the City who each recently undertook a large information systems project. We also looked at information technology strategic plans and policies in the states of Oregon and Kansas, and the City of Kansas City, Missouri. We found many similarities among the literature and government practices. We condensed and summarized these practices for this report.

"Managing large information technology purchases and projects is a skill that must be mastered by government. Public officials engaged in such projects must also get used to the idea that they must take risks to succeed."

Conference Report on Managing Technology, ***Governing 2002***

This study was conducted in accordance with generally accepted government auditing standards, including organizational independence, qualified staff, due professional care, and quality control.

Because many of the factors that lead to successful IS projects are managerial rather than technical in nature, the next section discusses the four major phases of IS development and the six major roles in IS project management. Twenty-three best practices in context of the four typical IS project phases follows. We have also included several appendices which provide more hands-on detail for IS project management. Throughout the report we reproduce quotes or ideas that we thought were particularly instructive. A list of the sources quoted is provided at the end of the report.

"Most successful projects utilize similar patterns of planning, estimating, and quality control technologies. . .

Poor project management tends to be the primary root cause for many failures."

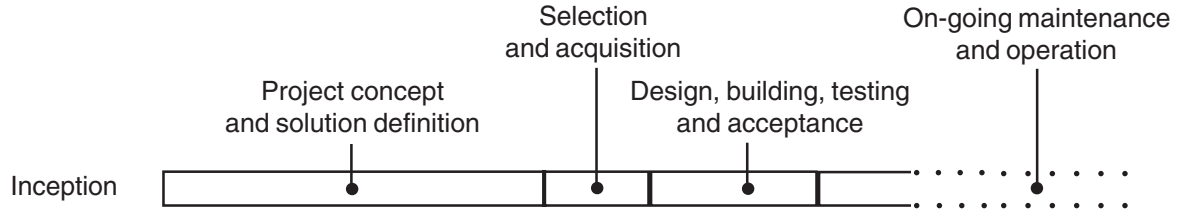
Patterns of Software Systems Failure and Success

Overview of phases and roles

From our research we learned that typically there are four distinct phases in the implementation of information systems, and six major roles for groups and individuals within those phases. Figure 1 shows the four phases along a conceptual time line. Best practices literature indicates that more time spent in project concept and solution definition pays off in quicker and more successful project implementation during the third phase.

The phases shown below apply to software that is purchased and modified for a specific information system application. However, software may also be developed by in-house staff or outside consultants. In these situations, the best practices pertaining to selection and acquisition may not be applicable.

Figure 1
INFORMATION SYSTEMS LIFE CYCLE PHASES



SOURCE: Audit Services synthesis of literature on information systems project management.

IS projects cycle through four distinct phases

Project concept and solution definition

This phase is largely a research and analysis phase where a need is identified and various stakeholders come together to decide whether or not to proceed with an IS solution. If the stakeholders decide to move ahead, functions of the proposed solution are evaluated against the organization's business strategy to discern the solution's compatibility with the organization's business vision and technological direction. Detailed project specifications are created. The stakeholders may decide to build the application in-house, find and customize existing software, or to outsource the entire function.

Selection and acquisition

In this phase a project team forms and clarifies the project with a detailed request for proposal. If the application is to be built in-house, the request for proposal may simply be a well constructed scope of work. A list of potential vendors is winnowed through RFP evaluation and site visits. Contract negotiations begin with the finalist, a contract is signed, and the project commences.

Design, building, testing and acceptance

In this phase the system is built or adapted to the organization's operating and technical environment. This phase is highly iterative, with predetermined deliverables being measured against a budget, time line and quality standards. Project meetings are frequent and communication and documentation are extensive.

On-going maintenance and operation

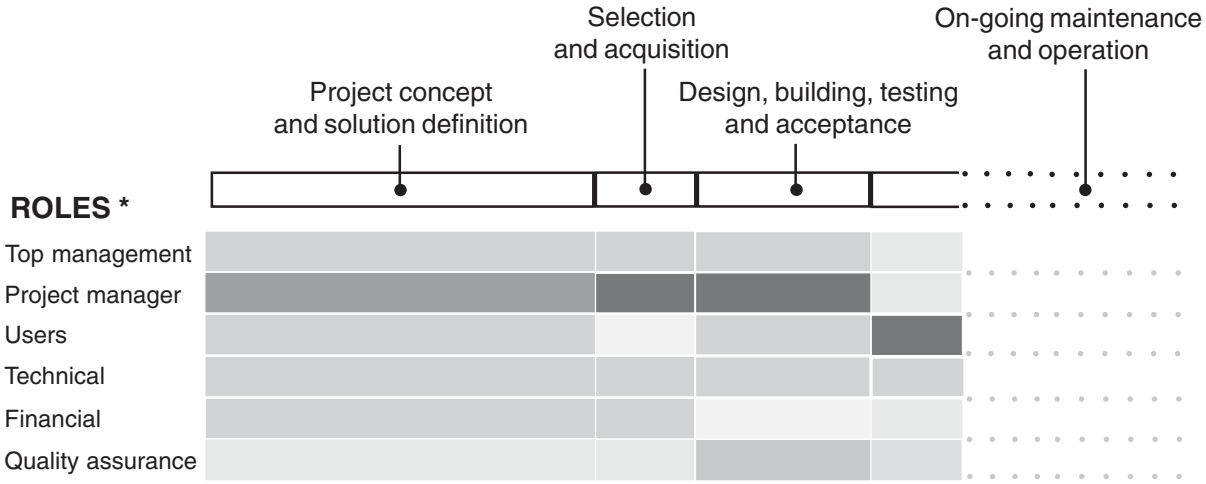
In this phase, the system is operational and maintenance and user training plans are carried-out. This phase also includes a post-implementation review that assesses the strengths and weaknesses of the entire project process. Over time, changes in software

versions and business processes will require additional changes and adaptations to existing systems. Eventually, the project life cycle begins anew.

IS projects require six distinct roles

The IS literature we reviewed consistently emphasized that people are the most important resource of an IS project. Successful IS development requires various perspectives and skills within the organization. Individuals, or groups of individuals, with specific skill sets and interests play distinct and synergistic roles throughout the project. All of these roles must be present for a successful project. As Figure 2 illustrates, some roles are more active than others during different phases.

Figure 2
KEY ROLES in Information Systems Life Cycle Phases



* Shading represents levels of involvement at each life cycle phase, with the darkest color being most involved.

SOURCE: Audit Services synthesis of literature on information systems project management.

Top management

Successful IS projects have top management that is fully supportive of, engaged in, and knowledgeable about the project. An individual, such as a bureau director or principal manager, must be ultimately accountable for the project. This role is also referred to as the project or executive sponsor. The project sponsor's responsibilities include the following:

- Ensuring that the project team keeps focused on the organization's business strategy and core business.
- Selecting and overseeing the project manager.
- Ensuring the project manager submits timely project status reports.
- Resolving crucial issues that the project manager cannot resolve.
- Providing final approval for major project changes, and if additional resources for those changes are necessary, procuring those resources.
- Taking full responsibility for the project if it fails.

"If not doomed from the start, an IT project without an executive sponsor – ongoing senior management – sets out on wobbly footing."

Contract Hoops and Loopholes, ***Governing*** magazine

Project manager

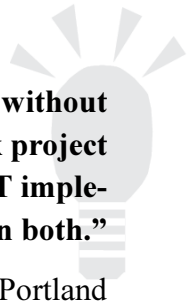
Project management is the linchpin of a successful IS project. The project manager is responsible for the schedule, budget, functionality, risk management, and overall implementation of the project. Project managers must be proficient in understanding and communicating both the technology of the project and the business concerns of the organization.

Project management is increasingly viewed as a professional discipline, and organizations and universities provide classes, workshops and training in project management. The State of

Kansas mandates training and certification of IS project managers because it recognizes the importance of sound management techniques for its IS projects.

The project manager's responsibilities include:

- Leading the project team through each phase of the project life cycle.
- Implementing and maintaining the project plan (e.g. GANTT and/or PERT charts).
- Managing changes to the project.
- Ensuring that the project budget, schedule and milestones are on track, and making changes if appropriate.
- Monitoring and communicating project risks, changes and status to top management.
- Getting approval for major changes from top management.
- Communicating all aspects of the project in different formats and to different types of stakeholders.
- Continually observing the attitudes of the project team in order to identify potential conflicts or differences of opinion and vision.



"Never undertake a major software implementation without experienced project management skills. Complex project management experience is critical. Experience in IT implementations is highly desirable. Insist on both."

draft *Systems Development Methodology,* City of Portland

Users

A person or group of persons who represent the users of the system can describe the business processes the system needs to perform. This task includes specifying the outputs of the system – such as reports used by middle managers for financial reporting. Users include end-users involved in day-to-day data entry, middle managers who rely on information for business decisions and reporting and, with today's increasing use of internet technologies, may also include individuals from outside the organization such as vendors and customers. Finally, just as it is important to have backing of the project from top management, it is important for users to be involved so they too can champion the project.

"Seek out a few key users. . . They can help communicate with other users, adopt the new technology early and review the system for you."

Riding the Tiger

Technical

Technicians interact with users, management and financial staff to determine the best technology, databases and system structure needed to make the project feasible and affordable. Technical experts evaluate system structure, maintenance requirements, and integration and compatibility with existing technology. In addition, technical experts can help translate user needs into technological requirements. Technicians are also needed during the testing phases of system implementation and for system maintenance and software upgrades.

Technicians offer expertise in a variety of areas such as:

- Helping users translate business requirements into technological functions.
- "Sizing" the proposed project – which means determining system capacity based on the amount of data that is to be processed, stored, and retrieved.
- Requiring the vendor to make specific modifications to integrate new third-party or existing software.
- Analyzing database requirements.
- Determining network, bandwidth, backup, memory, and system availability requirements.
- Determining hardware requirements.
- Ensuring that installation activities occur in proper sequence.
- Ensuring that the proposed system is compatible with existing technology and technological approaches within the organization.
- Converting existing data to the new system.
- Developing and checking the results of system, integration, and acceptance tests.

"However comprehensive a new system seems, there are always loose ends, unresolved interface issues, and functionality gaps."

The Smart Way to Buy Information Technology

Financial

Although top management may believe it aligns with business strategy, and users may love the new technology, a project must be financially justified. Financial staff must quantify the proposed benefits of a project and weigh them against the proposed costs. Financial staff are also valuable during vendor selection when a vendor's financial solvency is considered, and during contract negotiations when trade-offs that reduce, or increase, an organization's risk must be evaluated and reflected in the vendor's price. Financial staff should also be consulted when specifying an information system's financial reporting requirements.

Quality assurance

The use of an independent evaluator throughout all phases of a major information systems project is becoming increasingly common. For example, the State of Oregon requires that major system development projects (projects over $500,000) must set aside at least 4 percent of the project's funds for quarterly reviews by independent commercial evaluators. These reviews are summarized and reported to top management throughout each project stage.

Quality assurance checks the project's processes and the products of those processes. The quality assurance function should report on a regular basis to top management, the project manager, and any other oversight entity as appropriate.

Responsibilities for the Quality Assurance role may include:

- Determining that policies and plans are in place (e.g. change request orders, project plans, work breakdown structures).
- Monitoring that changes in scope are formally documented, processed, and approved.
- Ensuring that potential risks are continually identified and communicated.
- Monitoring the receipt of deliverables stated in the contract.
- Ensuring that problems that arise are documented, and top management, the project manager and the vendor are notified as appropriate.
- Ascertaining the quality of data that goes into, and comes out of, the testing environment.
- Ensuring that project documentation is kept up to date.
- Ensuring that training and maintenance plans are sufficient.
- Participating in peer reviews and the post-implementation review.

"[Someone needs to be in] a position to say what needs to be said without fear of reprisals or of looking bad when they bring problems to an executive sponsor. Someone has to be in a position to say 'Stop this before it gets worse.' "

Contract Hoops and Loopholes, ***Governing*** magazine

BEST PRACTICES

Within each phase of the project cycle for information systems software are key activities that contribute to the success of the overall project. Professional literature suggests "best practices" have evolved from experience and lessons learned.

The following sections describe twenty-three common best practices. They are presented chronologically, within the four life cycle phases.

A list of these best practices can be found in Appendix A.

Phase 1:
Project Concept and Solution Definition

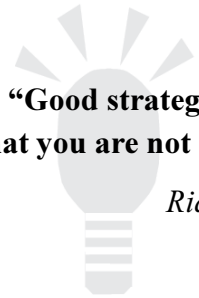
This phase is largely a research and analysis phase where a need is identified and various stakeholders come together to decide whether or not to proceed with an IS solution. If the stakeholders decide to move ahead, functions of the proposed solution are evaluated against the organization's business strategy to discern the solution's compatibility with the organization's business vision and technological direction. Detailed project specifications are created. The stakeholders may decide to build the application in-house, find or customize existing software, or to outsource the entire function.

BEST PRACTICES

1	Determine if you have a project.
2	Try to keep projects small and modular.
3	Describe the project in functional terms.
4	Know your business processes.
5	Evaluate the project financially.
6	Use project deliverables to define success.
7	Involve users early and often.
8	If you need outside help, get it.

1. Determine if you have a project.

Needs assessment is not only deciding what you need, but ***if*** you need it in the first place. We learned that ***strategic business planning*** should drive the development and design of information systems. For example, if an outdated computing system no longer supports a certain business function, it is wise to ask how important that business function is to the organization. Likewise, a technological advancement may sound appealing, but the organization must ask if adopting that advancement aligns with its strategic business decisions. If the problem is short term, or the service needed is not part of the organization's core business, the organization should consider outsourcing the function. Finally, some projects may be better suited to in-house development, rather than modification of purchased software.



"Good strategies are about what you are not going to do."

Riding the Tiger

Updates to information systems should enhance at least one of the following:

- basic operating tasks,
- quality to customers, or
- internal decision making.

After a project has been identified, desired features within a system should be evaluated against these three categories. In Appendix B, we present a series of questions to ask when an information technology project is proposed.

2. Try to keep projects small and modular.

In 1998 The Standish Group, which researches IS project management, found that most successful projects take no longer than six months and cost no more than $750,000. Projects costing $750,000 or less had a 55% success rate compared to 8% for projects ranging from $6 to $10 million.

In addition, creating an IS application architecture that uses layers of smaller software modules allows one to build systems in pieces that can be deployed and evaluated faster than a large project with dependent pieces. Additionally, use of common operating systems and standards facilitates the development of reusable software components and eases integration with other systems.

Finally, design for current business needs and try to design flexibility into the product. Products that are table driven may have more flexibility to meet future business demands without needing major rewrite or design.

"Any work that can be done beforehand to communicate and help a vendor understand your business processes is money in the pocket of the City."

Ann Kohler, Bureau of Development Services (paraphrased)

3. Describe the project in functional terms.

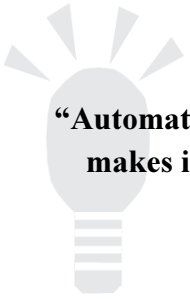
After deciding that there is a need for an IS project, the next step is to set out defining what the system should do. High-level business requirements should flow from business goals and users' needs.

To describe the project in functional terms, current workload data should be collected and compared to anticipated future conditions. These data include information on number of system users, number of transactions processed in a day, and the type of information recorded on those transactions. Important questions to ask are "what are the sources of the input data?", "what are the expected outputs?", "who is going to use this system and the information it provides?"

For example, a requirement could be "the system should be able to store all transactions per customer. Transactions will include A, B, C, D and E. In addition, summary customer information should be electronically retrievable in 5 seconds. Summary customer information will include X, Y and Z."

After the list is developed, you must decide what is absolutely needed, what would be nice to have, and what can be left out. It may be helpful to test these requirements against the three enhancement categories - basic operation, quality service or product, and organizational decision making - when ranking requirements. This ranking will be useful when developing, and comparing responses to, a request for proposal.

4. Know your business processes.



"Automating a bad process makes it more expensive. Fix it first."

Riding the Tiger

Seek to understand and document the businesses practices that will be supported by the proposed information system before writing a request for proposal and embarking on the vendor selection process. Although best practice literature refers to "reengineering business practices" to fit the new technology, current business practices must be understood first. The process of documenting current business procedures also produces the added benefit of uncovering and solving inconsistencies among those processes. Since inconsistencies often result in error, correcting for errors before automating them reduces the risk that the information system will produce wrong information.

5. Evaluate the project financially.

The literature we reviewed underscored the importance of identifying costs and benefits of the proposed project. In addition to the cost of the software license and installation, costs include project management, training, maintenance and hardware. Benefits include quantifying the magnitude of the existing problem, and the savings or increased revenue of the solution. Although a cost/benefit analysis is often difficult, it is needed in order to evaluate if the project is worth doing. Analysis can also help identify which system functions are critical and which are dispensable.

"Strategy is about selectivity. All managers know that some investments of their personal time, of their department's unit, pay back more than others do."

Riding the Tiger

One City information technology expert emphasized the importance of quantifying the *good will* benefit that government projects often produce. For example, IS projects may offer more information to citizens but not save money or result in increased revenues.

Appendix C lists potential costs and benefits to consider when evaluating an information system. Costs and benefits should be considered over a reasonable life-cycle for the system.

6. Use project deliverables to define success.

Defining the criteria for project success helps the project stay on track, clearly communicates project goals, and provides benchmarks against which to evaluate project performance. Project deliverables include major system performance requirements and project deliverables such as a communication plan, a scope of work, and a process to request and incorporate changes into the original scope of work.

Defining major system deliverables requires a great deal of detail. For example, the statement "The new billing system will send bills to all our customers and record payments received" must also address characteristics of billing and receipts such as accuracy, timeliness, amount of customer and payment detail captured, and level of detail reported in summary reports.

"Benchmarks of quality and performance of the project should be established at your earliest planning sessions."

IS Project Management Handbook

Additionally, system specifications could also state how the system is expected to handle unusual situations such as odd, heavy or repetitive data. The specifications should also define system availability, disaster recovery, and business continuity requirements – such as how long a business can manage without the system and how much data it can afford to lose.

Project deliverables are simpler to define and many may be transferable from one project to another. Appendix D provides a sample list of project deliverables.

7. Involve users early and often.

"...the only reason the project exists in the first place is because someone needs to use the business application when it is finished."

Unfinished Voyages,
The Standish Group International

An information system project must be used to be useful. Best practice literature emphasizes that users should be involved from the start of a project. In addition, there should be opportunities for users to provide input and review on user interfaces as they develop. Involving users in software design helps to ensure that business practices are consistent because design discussions may uncover inconsistent business practices. In addition, user involvement during design reduces the risk that users will develop their own personalized "work around" solutions to accommodate their needs. Overall, user involvement creates knowledge and ownership of the application.

8. If you need outside help, get it.

Some organizations do not have the skills or resources to successfully navigate through the information systems development process. If an organization does not have the personnel, experience or time to examine current business processes and functions in order to proceed with writing a request for proposal and managing a project, it makes sense to hire an outside consultant and/or project manager. External resources can also add value by challenging assumptions and asking "dumb" questions such as "why do you do it that way?", "what makes you think you can do it that fast?".

"When projects have problems, the cause is frequently an assumption that turned out to be invalid, or a constraint that was never identified."

Information Systems Project Management

Phase 2:
Selection and Acquisition

In this phase a project team forms and clarifies the project with a detailed request for proposal. If the application is to be built in-house, the request for proposal may simply be a well constructed scope of work. A list of potential vendors is winnowed through RFP evaluation and site visits. Contract negotiations begin with the finalist, a contract is signed, and the project commences.

BEST PRACTICES

9	Write a detailed and clear request for proposal.
10	Make it a team effort but speak with one voice.
11	Select two or three top finalists and see working versions of their software.
12	Negotiate a good contract.
13	Don't be afraid to stop and reevaluate if things are not going well.

9. Write a detailed and clear request for proposal.

Along with the software contract, the request for proposal (RFP) is the most important document in software acquisition. A thorough needs assessment lays the ground work for writing a good RFP. Functional statements and deliverables can be turned into system performance requests. The purpose of the RFP is to elicit meaningful, clear, and comparable vendor responses.

"An RFP imposes a disciplined process for identifying needs.

It forces vendors to formally make representations and commitments concerning the way they will meet those needs.

It helps a company think through its request and obtain valuable information and commitment at the point in the process when its leverage is strongest."

The Smart Way to Buy Information Technology

Therefore, RFP questions must be tightly written to avoid vague and ambiguous responses. Definitions should be used to help explain terms like "response-time" and "accuracy." In addition, limiting vendor responses to the following three options prevents the prospective vendors from answering "maybe" or "our system can do it all, with modification":

1. YES, the current system can absolutely perform this function and is included in the base price.
2. NO, the current system cannot do this and therefore is not included in the base price.
3. This function could be coded into the system but this modification would require additional resources beyond the base price of the software. This modification would cost between $_____ and $_____.

RFPs must also clearly communicate the organization's:

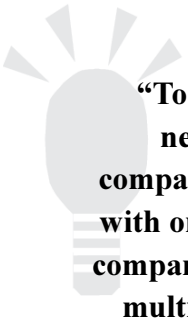
- current technical and operating environment,
- current technical and operating problems and desired solutions,
- organizational goals and vision, and
- roles and responsibilities.

RFPs should request information on:

- vendor financial information, organizational structure and personnel, industry position, industry references,
- the vendor's proposed implementation time line, the personnel assigned to the project, transition plans, software documentation, pricing, and payment terms,
- roles and responsibilities of the vendor, and if applicable, the vendor's partners, and
- the vendor's annual price for maintenance, upgrades and support (see best practice #21).

The Bureau of Purchasing has guidelines on writing RFPs and for contracting for Professional Technical Expertise (PTE). In addition, bureaus who have written RFPs and contracted licensed software applications may also offer guidance in the RFP and contracting process. Two good recent examples of RFPs are the City Auditor's e-files/records management, and the Bureau of Development Services permit-processing software requests.

Additionally, an informal request for information (RFI) could precede the RFP. An RFI may help the organization get a quick understanding of potential vendors in the market, and obtain information that will help the project team describe its needs. For example, the Bureau of Development Services invited vendors to present their software to a few potential users. These presentations helped staff become better consumers because new ideas for system functions were stimulated, and differences between software options were discussed.



"To be effective in negotiations, the company must speak with one voice. If the company speaks with multiple voices, the vendor will divide and conquer."

The Smart Way to Buy Information Technology

10. Make it a team effort but speak with one voice.

Successful information systems projects depend upon a team effort that includes top leadership, a skilled project manager, system users, financial analysts, technical expertise, and administrative oversight. However, the number of staff involved in the process can create opportunities for vendors to isolate team members and obtain internal confidential information or attempt to build inside alliances. To avoid these traps, designate a lead representative or negotiator, and decide what is and is not going to be said to potential vendors. The team should agree to external communication rules, such as not engaging in side conversations with potential vendors, avoiding oral promises, and keeping internal conflicts private.

11. Select two or three top finalists and see working versions of their software.

Limiting the finalists keeps the process competitive while not overburdening the selection team. A weighted attribute methodology is often used to select the top two or three software proposals. Software consultants such as META and GARTNER have developed weighting recommendations and evaluation criteria for software products.

If the team cannot winnow the list to two or three top proposals, they should reconsider the vendor's industry reputation, financial strength, personnel resources, and experience developing systems for similar organizations and operations.

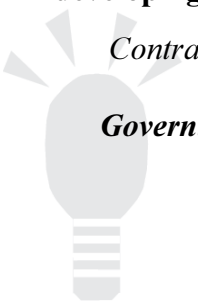
Once the finalists are selected, it makes sense to see working implementations of the software by performing site visits. If possible, team members should try to talk to customers out of earshot of the vendor to find out how they like the software and the vendor.

12. Negotiate a good contract.

Negotiating a good contract is key to fairly sharing risk between the organization and the vendor. Ideally the organization wants the vendor to take on as much risk as possible without the vendor walking away from the project. However, considering that an organization's business operations could fail if the software is inoperable, the organization always takes a greater risk than the vendor.

"To develop an effective negotiating team, governments need to launch a comprehensive effort to recruit representatives from all over the organization - from finance to technology to procurement to legal to the ultimate users of technology. Once in place, the negotiating team needs to understand the trade-offs that take place in developing a contract."

Contract Hoops and Loopholes, ***Governing*** magazine



Although an organization's negotiating leverage is affected by the number of qualified vendors in the marketplace and the value of the contract, a client that fails to develop a good negotiating team will greatly diminish any leverage it has.

Key points to remember when negotiating a contract include:

- Create a team that is representative of different organizational interests.
- Get the negotiation team to agree on the top objectives for negotiation.
- Speak with one voice (see best practice #10).
- If the team believes it is making too many concessions, take a break from negotiations or, suspend negotiations with the vendor and pursue other vendors on the short list.
- Build in "escape" mechanisms that allow for course correction and project termination.
- Link vendor compensation to successfully completed deliverables and milestones.
- Be familiar with software licensing terms and provisions.

Key terms and provisions are described in Appendix E.

13. Don't be afraid to stop and reevaluate if things are not going well.

Some of the literature we reviewed underscored the importance of being able to take a "time-out" if vendor negotiations, or any phase of system design and implementation, are not going well. Too many compromises during contract negotiations, missed milestones or errors during testing are signals to the project team that it may need to step back and reevaluate the project. Solutions to problems may be found, additional expertise may be brought in, or it might be appropriate to terminate the effort.

Strong leadership and quality assurance – two roles discussed in the beginning of this report – are critical support roles for this best practice. The project team may have too much invested to recognize a major problem. Strong leadership together with an objective reviewer may get a stalled project on its feet again, or support termination decisions.

"In addition to a politically savvy sponsor or committee, an IT project also needs independent oversight, whether through a contract arrangement or from a government entity that has no vested interest in the project."

Contract Hoops and Loopholes,
Governing magazine

Phase 3:
Design, Building, Testing and Acceptance

In this phase the system is built or adapted to the organization's operating and technical environment. This phase is highly iterative, with predetermined deliverables being measured against a budget, time line and quality standards. Project meetings are frequent and communication and documentation are extensive.

BEST PRACTICES

14	Have good systems in place for communicating, and monitoring deliverables and milestones.
15	Create a process for managing scope change requests.
16	Keep risks visible and managed.
17	"Chunk it" and clearly define end points.
18	Insist on thorough system documentation.
19	Test.
20	Prepare a system implementation plan early.

14. Have good systems in place for communicating, and monitoring deliverables and milestones.

"The core indicators of project health or dysfunction should be made readily available to all project participants. Anonymous channel feedback should be encouraged to enable bad news to move up and down the project hierarchy."

Software Acquisition Best Practices Initiative,
U.S. Department of Defense

The central function of project management is managing people and resources to ensure that deliverables arrive as expected, costs are controlled, and quality standards are maintained. In the previous section, we discuss the Project Manager's role in greater detail. Here we focus on the three tools to help the project stay on track: a communication policy, a single repository for project information, and an activity sequencing system.

Timely and appropriate communication is critical to ensuring that information flows among and between team members, is accessible, and is documented. A *communication policy* should outline who receives what types of information, when information is sent, how that information is documented and retrieved, and how problems are communicated to management and the vendor.

In addition, a *single repository of information* should be used for all data and actions relevant to the project. A server directory or web site is appropriate to store project information, such as:

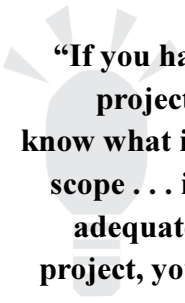
- actual to proposed project time line and milestones;
- actual to proposed budget (which includes the vendor's base cost as well as the organization's development and training costs);
- project team organizational chart(s);
- team meeting notes;
- testing results;
- change process requests; and
- open project issues.

Finally, an understanding of *activity sequencing* is key to project management. GANTT and PERT charts are often used to graph activities and their milestones. GANTT charts illustrate project status against a time line, but they do not represent task dependencies very well. PERT diagrams show how one task affects another. Hence, the two tools complement one another. However, GANTT charts can also be coded to indicate *critical path* activities (those activities that cannot slip without affecting the overall project schedule.)

"If you don't monitor, communicate, and follow-up, your rights under the contract can evaporate."

The Smart Way To Buy Information Technology

Project management software is available to produce and update schedules. Although its use is recommended in IS management literature, these software packages differ in their complexity and can be cumbersome to use. Some project managers who started using project management software eventually reverted to spreadsheet-produced schedules. Regardless of the project tracking tool, the project manager must have a method to schedule, track, monitor and communicate project activities.



"If you have defined the project properly, you know what is in and out of scope . . . if you have not adequately defined the project, you will now pay the price."

Information Systems Project Management

15. Create a process for managing scope change requests.

As a project develops, new opportunities and requests for system customization arise. Since scope expansions most always result in schedule slippages and added costs, these requests must be carefully managed.

A formal process, or at minimum a set of guidelines, should be established to identify and approve justified scope changes. The process should include or consider the following:

- The project vision should always be "in the face" of the project team - visible through slogans, icons, banners.
- Regular, separate meetings should be held to discuss requested scope changes.
- Only changes that are relevant to the project's core objectives should be considered.
- How the proposed change increases project risks.
- A scope change's potential benefits should be evaluated against its potential costs - including additional project management, quality control, and documentation costs.
- If a scope change is approved, resources dedicated to the project should be allocated accordingly.
- A change request form which includes the following information should be used:
    - date of request;
    - team member making request;
    - description of change;
    - reason for change;
    - relevance to project's core objectives;
    - estimated costs;
    - impact on schedule;
    - risk statement;
    - potential impact on other modules or systems;
    - date approved or rejected; and
    - authorized signature(s).

16. Keep risks visible and managed.

It would be difficult to find any information systems management literature that does not mention risk. Keeping risks visible allows the project team to continually reevaluate what mitigation plans, if any, are necessary. Best practices literature suggests that risk management should be a continuous activity throughout the project. Risks should be identified, categorized, ranked and reported as part of project team meetings and staff reports. Typical risks faced by IS projects include equipment failures, technological integration problems, deliverables not arriving as scheduled, changes in scope that are not approved by project management, or simply that staff is not available as, or is needed more than, expected.

". . . to manage risks, you must ensure that they are an overt part of the project team's consciousness."

Information Systems Project Management

"By utilizing quality gates, which prevent effort outputs to move on until they pass all their predefined acceptance criteria....(it's either done or it's not), evaluation of how actual progress is being made against the plan becomes meaningful."

Software Acquisition Best Practices Initiative,
U.S. Department of Defense

17. "Chunk it" and clearly define end points.

Dividing a large project into several small chunks, and assuring the quality of each chunk, before moving to the next phase ensures that value is derived from the project at the earliest possible stage. Activities in that stage should be closed only when performance standards are met.

Vendor payments should be linked to satisfactory chunk completion. In addition, software modules built as independent pieces can be re-used in other applications (see best practice #2).

18. Insist on thorough documentation.

Thorough project documentation is implicit throughout project management literature. In addition to the obvious need for documenting what the system does and how it does it, the assumptions and reasons behind the decisions made during system development should also be documented. Testing results must be documented.

19. Test.

Testing looks for errors during different phases of IS development, and is central to ensuring an operational product that does not require exorbitant support and maintenance costs once implemented. There are various types of software testing but four are most critical – functional, system, integration, and acceptance. Testing types are described in Appendix F.

The following best practices apply to testing:

- Budget for testing – this includes funds for developing the testing design and an adequate testing environment.
- Integrate testers into the project team at the start – **not as** endpoint inspectors.
- Test components as they are built or delivered.
- Allow time for testing, locating the source of errors, and error correction.
- Test with real data and real operators, and be aware of what data or scenarios are not included in the testing environment.

- Intentionally use bad data to test how the system responds.
- Track the defects found at different testing stages.

At some point the project manager will decide that acceptance testing is complete and will move the system into production. Critical elements that acceptance testing verifies include:

- the system is stable;
- all data is presented correctly – from data entry screens to final summary reports;
- the system interacts correctly with other systems and hardware;
- the data tables and relationships correctly represent the business transactions processed;
- system documentation is complete, and users are adequately trained;
- help desk support is available;
- system security and system administration functions are supported;
- batch processing works within the organization's sequencing and time requirements; and
- the system supports concurrent users and multiple transactions without delays or failure.

"The only way to keep program costs from exploding is by finding and fixing defects as they occur. The cost of fixing defects typically increases by a factor of ten as they pass into each subsequent development phase."

Software Acquisition Best Practices Initiative,
U.S. Department of Defense

20. Prepare a system implementation plan early.

The "live" deployment of the information system into the business environment is complex and requires close coordination among the project team, users and management. Implementation can occur while running the old system concurrently as a safeguard (parallel), or in controlled roll-outs focused on specific modules or user groups (phased), or conversion can occur all at once (full-blown). Consideration of how and when to implement the new system must weigh the pressures on the organization to "go-live" with the risks to the organization if the new system fails. Questions to consider include: How much data can the organization afford to lose? How long can critical operations be shut down if the system is inoperable? Are some days and dates better than others to cut-over to the new system (like over a holiday period or the beginning of a new financial reporting period)?

There are a variety of options that a project manager could consider when implementing [the system]. There are advantages and disadvantages to each type, and the choice usually depends on the client organizational setup and the complexity of [the system] to be implemented. . .experience has shown that developing a draft implementation schedule early in the project life, rather than later, resolves many problems. . .There is no turning back!"

Project Management Nation

Phase 4:
On-going Maintenance and Operation

In this phase the system is operational and maintenance and user training plans are carried-out. This phase also includes a post-implementation review that assesses the strengths and weaknesses of the entire project process. Over time, changes in software versions and business processes will require additional changes and adaptations to existing systems. Eventually, the project life cycles begins anew.

BEST PRACTICES

21	Plan for maintenance.
22	Invest in training.
23	Conduct a post-implementation review.

21. Plan for maintenance

Inevitably, system bugs will be found, further system modifications may be necessary, and new software versions will be released. Maintenance and system upgrade fees for these fixes and enhancements can be costly – particularly if the application is purchased rather than built in-house.

"...[in your contract] you may want to set forth detailed time service standards for response to or resolution of reported problems based upon defined severity levels."

The Smart Way To Buy Information Technology

The literature and experts we consulted recommended that system maintenance and upgrades costs be considered during the acquisition phase. When negotiating upgrade fees, the organization should consider that numerous bugs may be found and substantial retraining may be needed. It may be useful to request discounts on upgrades because, as one bureau manager said "The client is essentially funding some of the developer's research and development when it buys a new version." Finally, management should allocate internal staff time and funds needed for system configuration, security, debugging, and training.

"Benefits come from exploitation by the system owners and users. Allow enough time to prepare and train them or they will not be able to exploit the system effectively."

Project Management Nation

22. Invest in training

As one researcher stated "it's a lot easier to get money from a city council to buy a new computer system than to teach people how to use it properly." A training plan is necessary not only for end-users, but also for technical training foroperational and

help desk staff. Operational staff should know basic setup and restoration functions; help desk staff should be adequately prepared for a deluge of questions when the system goes live. Additionally, new versions of software may require substantial retraining.

23. Conduct a post-implementation review

A review after the system has been implemented serves four objectives:

- To determine and summarize the actual project costs and existing risks, and if immediate benefits were realized.
- To determine if the organization has fully exploited the system by changing its businesses processes to fit automation.
- To help continue top management accountability for project performance, and identify areas where system improvements and modifications need to be made.
- To inform decision making on future IS projects.

"Most federal agencies accept that recently implemented systems are a *fait accompli* and move on from there. This point of view is contrary to the investment management philosophy of the entire IT portfolio. The primary tool to assess a project in best practice organizations is the post implementation review."

Evaluating Information Technology Investments,
Federal Office of Management and Budget

Specific inquiries in the post-implementation review may include:

- How complex was this project? What factors added to its complexity?
- How did system delivery compare to the initial time-line and budget?

- Were project objectives and benefits defined and measurable? Were they achieved?
- What were the top issues or difficulties the project team faced? How adequately were they addressed?
- What plans or strategies were particularly useful? Why?

Historical information on information system implementations can help provide base line data for future projects, and guide the development of best practice actions that can support future software installations.

APPENDICES

APPENDIX A

List of 23 best practices

1. Determine if you have a project.
2. Try to keep projects small and modular.
3. Describe the project in functional terms.
4. Know your business processes.
5. Evaluate the project financially.
6. Use project deliverables to define success.
7. Involve users early and often.
8. If you need outside help, get it.
9. Write a detailed and clear request for proposal.
10. Make it a team effort but speak with one voice.
11. Select two or three top finalists and see working versions of their software.
12. Negotiate a good contract.
13. Don't be afraid to stop and reevaluate if things are not going well.
14. Have good systems in place for communicating and monitoring deliverables and milestones.
15. Create a process for managing scope change requests.
16. Keep risks visible and managed.
17. "Chunk it" and clearly define end points.
18. Insist on thorough system documentation.
19. Test.
20. Prepare a system implementation plan early.
21. Plan for maintenance.
22. Invest in training.
23. Conduct a post-implementation review.

APPENDIX B

Key questions to ask when an IS project is proposed

Project concept and solution definition

- What is to be done?
- Why is it to be done? Why does it need to be done now?
- How will it be done? Who will do it?

Describe how the project fits into strategic plans and information management objectives

- How does this project align with the bureau's strategic plan?
- How does this project align with the bureau's information goals?
- How does this project align with the City's strategic plan and information management objectives?
- Do other bureaus perform similar tasks? Can information be shared among bureaus?
- Is this a task that could be outsourced?
- What other bureaus or divisions does this project impact?

Alternative approaches

- What are the consequences of not proceeding with this project?
- What other approaches have been considered?

Considerations about the future

- Upon what key technologies does this project depend, and what is the expected evolution of these technologies?
- How will users change their requirements and interactions with the business processes this project supports as a result of this project?
- What are the maintenance needs of this solution? Who will maintain it?

Risk management and financial considerations

- Who will provide executive leadership, and be ultimately responsible for the success of this project?
- Who will fill the roles of:
  - Project Manager
  - System architecture and technology expertise
  - Financial oversight
  - Procurement / negotiation specialist
  - Quality control and assurance
  - Customer / Users / Other stakeholders
- What would happen if the project completely failed?
- What are the estimated resources needed for this project?
- Who will authorize this project?

APPENDIX C

Costs (⊖) and benefits (⊕) to consider

COSTS to consider in a cost/benefit analysis, by project phases

Project concept and solution definition

- ⊖ staff time to conceptualize and define system requirements
- ⊖ costs related to work not performed because staff time is utilized on project concept (could include overtime or temporary staffing costs)
- ⊖ costs for external consulting

Selection and acquisition

- ⊖ costs for project management
- ⊖ staff time to write RFP and evaluate proposals
- ⊖ costs related to work not performed because staff time is utilized on selection and acquisition (could include overtime or temporary staffing costs)
- ⊖ costs for external consulting or contract analysis
- ⊖ travel costs for site visits

Design, building, testing, and acceptance

- ⊖ labor costs for project management
- ⊖ staff time related to project design, building, customization, testing and accepting
- ⊖ costs related to work not performed because staff time is utilized on project design, building, testing and accepting (could include overtime or temporary staffing costs)

(Design, building, testing, and acceptance, continued)

- capital costs for new, or upgraded hardware
- capital costs for new networking or communications equipment
- costs to configure and install hardware
- software licenses
- software installation and development costs
- costs to create staging environments for testing
- costs to integrate new system with existing systems
- costs to convert old data to new system
- costs to run concurrent systems before cutover and phaseout
- costs for upkeep of project management document repository
- loss on sale of old equipment

On-going maintenance and operation

- costs of hardware and software maintenance contracts
- cost for hardware and software upgrades
- costs related to staff training such as time, materials and facility costs (including lost productivity during training)
- severance costs if new system reduces staff

BENEFITS to consider in a cost/benefit analysis

Benefits are difficult to quantify because they are based on predicted, rather than certain, outcomes. Most benefits will be realized after the costs have been paid. A potential benefit should only be counted if it would not have happened if the new system was not installed. Possible benefits include:

- gain on the sale of old equipment
- reduction in labor costs for maintaining old hardware and supporting old software systems
- reduction in materials costs for maintaining old hardware
- decrease in personnel (e.g. personnel needed to maintain old system, or staff reduction due to productivity improvements)
- productivity improvements (e.g. quicker cash receipts, better service delivery, better inventory management, better decision making)
- increased revenue (e.g. through the ability to capture new customers because of new products or services offered)
- future cost avoidance and reduction in uncertainty or risk from maintaining a legacy system
- "good will" benefits

APPENDIX D

Checklist of project deliverables, by project phase

This list is a sample of deliverables that may apply to a project.

Project concept and solution definition

- ☐ Project vision statement.
- ☐ Statement of problem(s) to be solved.
- ☐ List of proposed system requirements justified by how the requirements enhance basic operating tasks, quality to customers and internal decision making. System requirements should address volume of information processed in addition to type, speed, desired output, format, etc.
- ☐ Alternatives analysis.
- ☐ Cost / benefit analysis.
- ☐ Project estimate.
- ☐ A list of indicators which details what constitutes project success. These indicators can quantify any number of things – including processing time, savings or revenue generated, types of information produced, or the degree to which customer or user satisfaction is enhanced. Indicators could also be used to describe the success of the project's process – such as speed at which internal conflicts are successfully resolved.

Project selection and acquisition

- ☐ RFP proposal and evaluation procedures.
- ☐ Acquisition and / or project team organizational chart.
- ☐ Communication policy and plan. This includes creating a repository for all project information, and policies for updating information and communicating problems to the vendor.
- ☐ Data dictionary.
- ☐ Definitions.

Design, building, testing and acceptance

- ☐ Application documentation.
- ☐ Data models.
- ☐ Business rules executed by the system.
- ☐ Change request and approval policy and form.
- ☐ Risk management policy and plan.
- ☐ Work breakdown structures.
- ☐ GANTT and/or PERT charts.
- ☐ Testing plans for different testing procedures (e.g. functional, system, integration, and acceptance.)
- ☐ Plans for cutover to new system, and phaseout of old system.
- ☐ Program code.

On-going maintenance and operation

- ☐ Maintenance plan.
- ☐ Training plan and materials.
- ☐ Post-implementation review plan.

APPENDIX E

Selected software licensing issues & contract provisions

The following are a selection of issues and provisions typically found in a software license contract. The primary source for these descriptions was *The Smart Way to Buy Information Technology* (see Appendix G). The descriptions are presented here to give the reader an understanding of the issues involved in a typical software license. This list is not exhaustive; other provisions and requirements apply that are not discussed here. Check with the City Attorney's Office for further clarification of these terms. The City Attorney's Office must sign and approve any PTE contract.

Acceptance and acceptance tests

States under what conditions the licensee will accept the software and pay the vendor.

comments:

Specify the criteria under which the software is acceptable, including acceptance test results. Link vendor payments to different phases of implementation. Withhold final payment till several months after the system is fully operational and performing as stated in the acceptance criteria.

Assignment

The assignment clause delineates to whom the licensee can, or cannot, transfer its contracts rights and responsibilities.

comments:

Agreeing to a prohibition on assigning the license could be costly if, in the future, the licensee decides to outsource the function the software provides.

Compensation payment

How the vendor gets paid for its services.

comments:

Payments should correspond with the timing of the deliverables stated in the scope of work. The contract should outline a process for resolving disputed payments.

Confidentiality

Helps protect the vendor from its competitor's gaining access to trade secrets, and/or its client disclosing software or documents relating to it.

comments:

Avoid making the terms of the contract itself confidential and clauses that "deputize" the licensee to watchdog the vendor's confidentiality. Be aware that confidentiality clauses will apply to any consultants or employees that have access to the vendor's confidential information.

Definitions

The meaning of terms used in the contract.

comments:

It is helpful to consolidate the definitions for the terminology used in the contract. It also helps the licensee think through what things like "on time" and "acceptable" and "satisfactory" mean, and to define business terms unique to the licensee's operations.

Entire agreement

States what documents are part of the contract.

comments:

The licensee should ensure that any promises or performance guarantees made during negotiations are included as part of the contract.

Grant of license

The grant of license provides the basic right to use the software as well as the what, by whom, for whom, for what, where, until when, and why of the software license.

comments:

The licensee must consider how it needs to use the software (e.g. by how many users and at what locations) and write the contract based on those needs.

Limitation of liability

The dollar limit, if any, for which the vendor is responsible for a variety of risks. These risks include direct damages and other types of losses.

comments:

Consult with the City Attorney when negotiating liability limits.

Scope of use

Defines assigned uses such as where the software can be used, what hardware can run it, how many people can use it.

comments:

The licensee should seek as unrestrictive a scope of use as possible. (But the licensee should never pay for additional capacity it does not currently need.)

Scope of work

The plan of action between the client and the vendor. A scope of work should clearly specify all items to be delivered, the timing of these items, and the quality standards for these items.

comments:

Take time to clearly specify your requirements. This protects both the licensee and the vendor.

Source code in escrow

The source code is strings of programming language that make up the software. Source code is needed to make major software modifications and is the primary "good" that the vendor sells. Although there is a growing body of software that is "open source" where the source code is provided, many vendors do not share their source code.

comments:

Vendors need to protect their source code and therefore usually do not make it available to the licensee. However, the licensee needs to gain access to the source code in the event that the vendor goes out of business and the licensee needs to modify its software application. Putting the source code in

source code in escrow, continued

escrow with an independent third party escrow agent is generally the way the licensee is protected.

Escrow clauses should contain all common ways that a vendor could fail such as bankruptcy, insolvency, or failure to maintain the software. The clauses should also state, and the licensee should periodically ensure, that the vendor continually deposits new releases and related technical documentation into the source code account.

Termination

Defines under what conditions the contract may be terminated.

comments:

The licensee should not allow the vendor to terminate the contract except in extreme circumstances where the licensee is not fulfilling its obligations (such as not paying the vendor as agreed.) The licensee should be able to terminate with the vendor if the vendor continues to breach contract deliverables such as delivery dates or performance standards. The termination clause should also outline provisions under which the vendor would refund monies already paid, and pay for damages caused by the breach of contract.

Warranty, obligation and responsibility

This provision details what the vendor's and the licensee's responsibilities are in carrying out the contract and supporting the software. The provision also details what defects will be covered, for how long they will be covered, and under what circumstances they will not be covered.

comments:

The licensee should pay careful attention to what it is obligated to do, and make sure that it follows through with those obligations. In cases where the licensee's failure to perform responsibilities results in the vendor's inability to meet its obligations, the licensee should limit its liability only to instances where the vendor's failure is directly related to the licensee's failure.

APPENDIX F
Types of testing

Descriptions of different testing types

There are various types of testing that occur in the development and installation of software. The project manager must verify which types of testing are applicable to the project, and establish appropriate scenarios and benchmarks for each testing type. A crucial component of testing is devising testing scenarios that closely approximate the end-user environment. Therefore, it is important to use real users, real data, and real data loads.

Unit testing is low-level detail testing to verify that application code is performing as expected. With licensed software, much of the unit testing has already occurred at the vendor's site by programmers. Code changes made to adapt the vendor's software to its client's operating and technical environment need to be unit tested.

Pilot testing checks the software's performance before it is deployed at the users' site. The software is tested in an environment that simulates how it will be used. Pilot testing is an iterative process that checks for errors in order to make subsequent software improvements. It is performed by the software developer.

Functional testing looks at specific technical or business requirements to ensure that the software functions as expected. Functional testing uses the software's interfaces to determine if "what you see is what you get." Quality assurance staff is usually responsible for the planning, documentation and communication of problems found during functional testing.

Integration testing determines that different modules and layers of the software work correctly together. This testing is extremely important to confirm that the system processes data reliably, consistently and correctly when accessed through different business interfaces (for example, through accounts receivables and order processing.)

Security testing tests how well the system is protected against unauthorized access and willful damages.

System testing, also called "end-to-end" testing, verifies that the system performs as a complete application in a staged environment that mimics the real-world environment. This means the system interacts with other databases, networks, and communication and hardware devices.

Stress testing literally stresses the system to see how it responds to unusually large data loads, numerical calculations, complex query requests or repetitions of selected actions.

Acceptance testing is the last type of testing to occur and confirms that the end-users are satisfied with the completed and installed system. Acceptance of the system is based on satisfaction with the system over some predetermined time frame.

APPENDIX G

Sources quoted

Charvat, Jason. 2002. ***Project management nation****: tools, techniques, and goals for the new and practicing IT project manager*. New York, NY: John Wiley and Sons.

Davidson, Alistair, Harvey Gellman and Mary Chung. 1999. ***Riding the tiger****: how to outsmart the computer that is after your job, how not to bankrupt your organization with information management, how good clients get exceptional results*. Indianapolis, IN: Park Avenue Productions.

Doss, George M. 2000. ***IS project management handbook***. Prentice Hall.

Federal Office of Management and Budget. 1995. ***Evaluating information technology investments***. http://www.whitehouse.gov/omb/inforeg/infotech.html

Governing. July 2002. ***Contract hoops and loopholes***.

Governing 2002. ***Conference report managing technology 2002****: policy, politics and leadership*. governing.com/mtech2cr.htm

Hallows, Jolyon. 1998. ***Information systems project management****: how to deliver function and value in information technology projects*. New York, NY: American Management Association.

Jones, Capers. 1995. ***Patterns of software systems failure and success***. Boston, MA: Thomson Computer Press.

Peterson, Brad L. and Diane M. Carco. 1998. ***The smart way to buy information technology****: how to maximize value and avoid costly pitfalls*. New York, NY: American Management Association.

The Standish Group International. 1999. ***Unfinished voyages***. The Standish Group. www.standishgroup.com/sample_research/unfinished_voyages_1.php

United States Department of Defense. 1995. ***Software acquisition best practices initiative.*** www.spmn.com

THIS REPORT IS INTENDED TO PROMOTE THE
BEST POSSIBLE MANAGEMENT OF PUBLIC RESOURCES

If you no longer need this copy, you may return it to the
Audit Services Division. We maintain an inventory of past audit reports
and your cooperation will help us save on printing costs.

Audit Services Division
City of Portland
1221 S.W. 4th Avenue, Room 310
Portland, Oregon 97204

The report is also available on the Audit Services Division's web page:
http://www.portlandonline.com/auditor

The web page version of the report is the same as the printed version,
and can be downloaded and printed from most laser printers.