SAMPLE INVITATION TO TENDER ADVERTISEMENT
(CONTRACT)

Invitation to Tender
[Insert brief description of project/consultancy –
E.g. “provision of legal services for X native title claim”].

[Name of Representative Body] is calling for tenders for [insert brief description of project/consultancy – e.g. “provision of legal services for X native title claim”].

Tender documentation, including terms of reference and selection criteria, can be obtained from:
[Insert name and title of person responsible]
[Insert Representative Body address]
[Insert fax no.]
[Insert phone no]

The tender closes on [insert closing date and time].

SAMPLE INVITATION TO TENDER ADVERTISEMENT
(CONSULTANT REGISTER)

INVITATION TO TENDER Establishment of Consultant Register

[Name of Representative Body] is establishing a Consultant Register which will cover the following fields of expertise:
  o [Insert list of fields of expertise for which Consultant Register is to be established]

Application forms and selection criteria can be obtained from:
[Insert name and title of person responsible]
[Insert Representative Body address]
[Insert fax no.]
[Insert phone no]

The tender closes on [insert closing date and time].

An Invitation to tender is a situation where a customer asks many different suppliers to respond to a bid for a specific request.

An invitation to tender is indeed a competitive environment. Both the customer and the supplier know that there will be a comparison of all the suppliers' products when the decision to pick the winning bid is to be decided upon. There are two very important techniques to use for this scenario. They are prototyping and QFD. Prototyping is well suited for this scenario because, with today's 'visual' languages, a prototype can be made very rapidly at low cost. The QFD technique is also important
Tender Analysis

Consultant quantity surveyors normally carry out tender evaluation. Generally, tenders shall be evaluated based on the following factors:

- **Price** - The tender price is compared to the consultant QS’ estimates, previous tender prices/unit rates and next two lowest tenderers’ prices to satisfy that the tender is competitive and prices fair and reasonable.
- **Compliance with tender specifications** - The tender should comply fully or very substantially with tender specifications.
- **Past performance** - The tenderer's past performance must be satisfactory.
- **Life cycle costing, maintenance cost, present value of money and currency fluctuations**, if applicable.

Generally, if everything is equal, price shall be the main consideration.

A recommended tender shall be checked for arithmetical and pricing errors. For public sector projects, the recommended tenderer's debarment and registration status shall also be checked.

On completion of evaluation, a Tender Evaluation Report is prepared, and signed by the consultant architect and quantity surveyor. It is sent to the client for approval. After the client’s approval is obtained, a letter of acceptance of tender is then issued to the successful tenderer.

In private projects, tender interviews may be carried out, to review prices. Negotiations are carried out among consultants, contractor, suppliers, and subcontractors. During these negotiation sessions, contractors should:

- Show keen interest and respond quickly
- Safeguard the company’s interest
- Be on a lookout for competitors’ actions.

Rent or build Decisions.

Some of the most important questions facing managers are “how should we acquire and maintain our technology assets? Should we build and run them ourselves or acquire them from outside sources?” In the past companies built and ran their own computer facilities and developed their own software. Today, more and more companies are obtaining their hardware and software technology from external service vendors. On-line services for storage and running application software have become especially attractive options for many firms.

On-line storage service providers

Some companies are using storage service providers (SSPs) to replace or supplement their own in-house storage infrastructure. A storage service provider is a third party provider that rents out storage space to subscribers over the Web. SSPs sell storage as a pay per use utility, allowing customers to store and access their data without having...
to purchase and maintain their own storage infrastructure and storage support staff. To be successful SSPs must offer very high availability and reliability and also must keep up with the latest technology. SSPs are responsible for monitoring the stored data and for managing their own capacity, response time and reliability.

**Application Service Providers (ASPs)**

An application service providers (ASPs) is a business that delivers and manages applications and computer services from remote computer centres to multiple users via the Internet or a private network. Instead of buying and installing software programs, subscribing companies can rent the same functions from these services. Users pay for the use of this software either on a subscription or per transaction basis. The ASPs solution combines package software applications and all of the related hardware, system software, network and other infrastructure services that the customer would have to purchase, integrate and manage his or her own. The ASP customer interacts with a single entity instead of an array of technologies and service vendors.

The “timesharing” services of the 1970s, which ran applications such as payroll on their computers for other companies were an earlier version of this application hosting. But today’s ASPs run a wider array of applications than these earlier services and deliver many of these software services over the web. At web-based services, servers perform the bulk of the processing and the only essential program needed by users is their web browser. Large and medium sized businesses are using them for functions such as invoicing, tax calculations, electronic calendars and accounting.

Companies are turning to this “software utility” model as an alternative to developing their own software. Some companies will find it much easier to “rent” software from another firm and avoid the expense and difficulty of installing, operating and maintaining complex systems such as enterprise resource planning (ERP). The ASP contracts guarantee a level of service and support to make sure that the software is available and working at all times. Today’s Internet driven business environment is changing so rapidly that getting a system up and running in 3 months instead of 6 could mean the difference between success and failure. ASPs also enable small and medium size companies to use applications that they otherwise could not afford.

Companies considering the software utility model need to carefully assess ASP costs and benefits, weighing all management, organisational and technology issues. In some cases the cost of renting the software can add up to more than purchasing and maintaining the application in house. Yet there may be benefits to paying more for software through an ASP if this decision allows the company to focus on core business issues instead of technology challenges.

**Evaluating Software Alternatives**

- **Make or buy decision**
  - Develop software in-house
  - Purchase a software package
  - Customize a software package
Outsourcing
End-user (or departmental) computing

Developing software in-house
Reasons for in-house development
1. Satisfy unique requirements
2. Minimize changes in business procedures and policies
3. Meet constraints of existing systems
4. Meet constraints of existing technology
5. Develop internal resources and capabilities

Buying a software package
Reasons for buying a software package
1. Lower costs
2. Requires less time to implement
3. Proven reliability and performance benchmarks
4. Implemented by other companies
5. Requires less technical development staff
6. Future upgrades provided by the vendor

Customizing software packages
1. Purchase a basic package that can be customized to suit your needs
2. Negotiate with software vendor to make enhancements to suit your needs
3. Purchase the package and make your own modifications

Outsourcing
1. Using outside companies to handle part of the workload, on short-term or long-term basis
2. Contract personnel firms
3. Systems management or facilities management firms

End-user systems
1. Major factor in systems planning and development
2. Applications can be managed by end-users
3. Software suites offer integrated applications
4. Interactive Help features include wizards
5. Security concerns might require read-only files
6. Information centres (IC) can support end-user systems

Steps in Evaluating and Purchasing Software Packages
Five step process
1. Evaluate the information system requirements
2. Identify potential software vendors
3. Evaluate software package alternatives
4. Make the purchase
5. Install the software package

Evaluate the information system requirements
1. Identify the key features of the system
- Estimate volume and future growth
- Specify any hardware constraints
- Prepare a request for proposal or quotation

**Identify potential software vendors**
- Next step is to contact potential vendors
- An RFP will help vendors to identify solutions
- Various sources of information on suppliers
  - Retailers
  - Computer manufacturers
  - Industry trade journals
  - Systems consultants

**Evaluate software package alternatives**
- Object is to compare software packages and select the best alternative
- Obtain information from many sources
  - Vendor presentations and literature
  - Product documentation
  - Trade publications
  - Companies that perform software testing/evaluation
  - Contact users of the package
  - Benchmark test

**Make the purchase**
- Software licenses
- Lease agreements
- Maintenance agreements

**Install the software package**
- Installation time depends on size and complexity
- Before using the package, complete all implementation steps
  - Loading, configuring, and testing the software
  - Training users
  - Converting data files to new format

**Hardware Alternatives**
Hardware decisions use the same five-step approach as software decisions
1. Evaluate system requirements
2. Identify potential hardware vendors
3. Evaluate hardware alternatives
4. Make the purchase
5. Install the hardware

**Other issues to consider**
- Site preparation
- New workstations
- Network cabling
- Raised floors
- Conditioned electrical lines
- Fire extinguishing equipment
o Uninterruptible power supplies (UPSs)

How do you select the best alternative?

Most companies’ combine
  o In-house developed software
  o Software packages
  o Outsourcing
  o End-user systems

Object is to develop a list of viable alternatives; all viable alternatives must be evaluated. Feedback from users is essential

Presentation to management

Presentation guidelines and suggestions
  1. Give overview of the project’s purpose and objectives
  2. Summarize alternatives, with costs, pros, and cons
  3. Explain why the recommended alternative was chosen
  4. Allow time for discussion, questions, and answers
  5. Obtain final decision from management or timetable for next step

Five probable management decisions
  1. Develop an in-house system
  2. Modify the current system
  3. Purchase or customize a software package
  4. Perform additional systems analysis work
  5. Stop all further work

Testing

Exhaustive and thorough testing must be conducted to ascertain whether the system produces the right results. Testing answers the question “Will the system produce the desired results under known conditions?”

The amount of time needed to answer this question has been traditionally under-rated in systems project planning. Testing is time consuming: Test data must be carefully prepared, results reviewed and corrections made to the system. In some instances parts of the system may have to be redesigned. The risks of glossing over this step are enormous.

Testing is necessary to ensure that all programs function correctly. First step is to detect syntax errors and obtain a clean compilation. Next step is to eliminate logic errors, techniques include desk checking, structured walkthrough, and code review. Final step is testing, testing an information system can be broken down into 3 types of activities,
  • Unit (module) testing
  • System testing
• Acceptance testing

**Unit testing**
Consists of testing each program separately in the system. It is widely believed that the purpose of such testing is to guarantee that programs are error free, but this goal is realistically impossible. Testing should be viewed instead as a means of locating errors in programs, focusing on finding all the ways to make a program fail. Once pinpointed problems can be corrected.

**System testing**
Tests the functioning of the information system as a whole. It tries to determine if discrete modules will function together as planned and whether discrepancies exist between the ways the system actually works and the way it was conceived. Among the areas examined are performance times, capacity for file storage and handling peak loads, recovery and restart capabilities and manual procedures.

**Acceptance testing**
Provides the final certification that the system is ready to be used in a production setting. System tests are evaluated by users and reviewed by management. When all parties are satisfied that the new system meets their standards the system is formally accepted for installation.

The systems development team works with users to devise a systematic test plan. The test plan includes all of the preparations for the series just described.

E.g. of a text plan.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Address and maintenance “Record Change Series”</th>
<th>Test Series 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test ref:</td>
<td>Condition tested</td>
<td>Special</td>
</tr>
<tr>
<td></td>
<td></td>
<td>requirements</td>
</tr>
<tr>
<td>2</td>
<td>Change records</td>
<td>Key field</td>
</tr>
<tr>
<td>2.1</td>
<td>Change existing records</td>
<td>Other fields</td>
</tr>
<tr>
<td>2.2</td>
<td>Change nonexistent record</td>
<td>Deleted fields</td>
</tr>
<tr>
<td>2.3</td>
<td>Change deleted record</td>
<td>record must be available</td>
</tr>
<tr>
<td>2.4</td>
<td>Make second record</td>
<td>Change 2.1 above</td>
</tr>
<tr>
<td>2.5</td>
<td>Insert record</td>
<td>OK if valid</td>
</tr>
<tr>
<td>2.6</td>
<td>Abort during change</td>
<td>Abort 2.5</td>
</tr>
</tbody>
</table>

How far should you go with system testing? Pressure for the new system from users and managers vs. the need to avoid major errors.
Typical issues to consider
- What is the judgment of analysts, programmers, IS management, and the project manager?
- Do potential problems exist that might affect the integrity or accuracy of data?
- Can minor changes be treated as future maintenance items?

Management Approval
After system testing is complete, the results are presented to management:
- Test results
- Status of all required documentation
- Input from users who participated
- Recommendations
- Detailed time schedules, cost estimates, and staffing requirements

Management Options
- Return to Design Phase
- Retest
- Proceed with Implementation

If approved, a schedule for system installation and evaluation will be established

Training
A training plan should be considered early in the systems development process. Deliver the right training to the right people at the right time
Specific training is necessary for
1. Users
2. Managers
3. IS department staff members

Vendor training
If hardware or software is purchased outside, vendor training should be considered. Many vendors offer free or nominal cost training for customers. Vendor training can be performed at the vendor’s site or at the customer’s location. Vendor training often provides the best return on training dollars.

Outside training resources
If vendor training or internal training is impractical, outside trainers or consultants can be used. Outside training generally is not practical for in-house developed systems. Many sources of training information exist:
- Consultants
- Universities
- Information management organizations
- Industry associations

In-house training
IS staff and user departments usually share responsibility for developing and conducting training for in-house systems. Training techniques can include many
techniques and training aids, including multimedia, demonstrations, videotapes, and charts

Some Training Guidelines to Consider
1. Train people in groups, with separate programs for distinct groups
2. Select the most effective place for training
3. Provide for learning by hearing, seeing, and doing
4. Prepare a training manual
5. Develop interactive tutorials and training tools
6. Rely on previous trainees
7. When training is complete, conduct a full-scale simulation for users to gain experience and confidence

File Conversion
File conversion can take place after the operational environment is established and training has been performed.
Issues to consider:
- Automated conversion techniques
- Methods of exporting data to the new system
- Programs designed to extract and convert data
- Controls required to protect vulnerable data
- Verification of results by users

Conversion
The process of changing from the old system to the new system. Four main conversion strategies
- Parallel
- Direct cutover
- Pilot study
- Phased approach
Each approach involves different cost and risk factors.

Parallel
Both the old system and its potential replacement are run together for a time until everyone is assured that the new one functions correctly. Data is input to both systems, and results can be verified. This is the safest conversion approach because in the event of errors or processing disruptions the old system can still be used as a backup. However this approach is very expensive and additional staff or resources may be required to run the extra system. This method is impractical if the systems are dissimilar or cannot be supported together

Direct cutover
Replaces the old system entirely with the new system on an appointed day. At first glance this strategy seems less costly than the parallel conversion strategy. However it is a very risky approach than can be potentially be more costly than parallel activities if serious problems with the new system are found. There is no other system to fall back on. Dislocations, disruptions and the cost of corrections may be
enormous. Timing is an important factor for systems that have periodic processing cycles

**Pilot study**
Introduces the new system to only a limited area of the organisation, such as a single department as operating unit, the rest of the organization continues to use the old system. When this pilot version is complete and working smoothly it is installed throughout the rest of the organisation, either simultaneously or in stages. Cost is relatively moderate, because only one location runs both systems. Risk also is relatively moderate, because the new system is installed only at the pilot site and the risk of failure is reduced.

**Phased approach**
Introduces the new system in stages either by functions or by organisational units. If for example the system is introduced by functions, a new payroll system might begin with hourly workers who are paid weekly followed 6 months later by adding salaried employees (who are paid monthly) to the system. If the system is introduces by organisational units corporate headquarters might be converted first followed by outlying operating units 4 months later. Cost is relatively moderate, because the system is implemented in stages, rather than all at once. Risk also is relatively moderate, because the risk is limited to the module being implemented.

Moving from an old system to a new one requires that end users can be trained to use the new system. Detailed documentation showing how the system works both from a technical and end user standpoint is finalised during conversion time for use in training and everyday operations. Lack of proper training and documentation contributes to system failure so this portion of the systems development process is very important.

**Documentation**
Explains the system and helps people interact with it.

**Types of documentation**
- Program documentation
- System documentation
- Operations documentation
- User documentation

**Program documentation**
 Begins in the systems analysis phase and continues during systems implementation. Includes process descriptions and report layouts. Programmers provide documentation with comments that make it easier to understand and maintain the program. An analyst must verify that program documentation is accurate and complete.

**System documentation**
System documentation describes the system’s functions and how they are implemented. Most system documentation is prepared during the systems analysis and systems design phases.
Documentation consists of
  o Data dictionary entries
  o Data flow diagrams
  o Screen layouts
  o Source documents
  o Initial systems request

Operations documentation
Typically used in a minicomputer or mainframe environment with centralized processing and batch job scheduling. Documentation tells the IS operations group how and when to run programs. Common example is a program run sheet, which contains information needed for processing and distributing output.

User documentation
Typically includes the following items
  o System overview
  o Source document description, with samples
  o Menu and data entry screens
  o Reports that are available, with samples
  o Security and audit trail information
  o Responsibility for input, output, processing
  o Procedures for handling changes/problems
  o Examples of exceptions and error situations
  o Frequently asked questions (FAQ)
  o Explanation of Help & updating the manual
  o Written documentation material often is provided in a user manual

Analysts prepare the material and users review it and participate in developing the manual. Online documentation can empower users and reduce the need for direct IS support
  Context-sensitive Help
  Interactive tutorials
  Hints and tips
  Hypertext
  On-screen demos