Acquiring Software

Whether purchased as shrink-wrapped software or contracted as a bespoke development, software acquisition needs to be carefully considered. The following checklists have been constructed over years of (sometimes painful) experience.

Scope

Does the contract align with company acquisition strategy?

General Contractual

Is there a clear definition of each parties roles and responsibilities e.g. RASIC (or RACI) (see appendix A)?

Is there a Statement Of Work (SOW) (see appendix B) that covers the work to be undertaken?

Does the supplier protect you against patent infringement or other legal challenges against the supplier's software?

Has an evaluation been carried out to identify use of Open Source software and licensing implications of the use of such software?

Project Management

Is there a risk management plan identified to guide the software development through to a successful conclusion? N.B. This is in addition to project management and control plans identified by the supplier.

Is there a Software Quality Assurance plan? Is it based on recognised standards e.g. IEEE 730? Or ISO/IEC 25051

Are Configuration Management requirements clearly defined?

Supplier Capability

Is there a clear definition of the required level of software capability and maturity (see appendix C) to be demonstrated by the supplier?

Has the supplier been audited to ensure that they meet the required level of capability and maturity?

Oversight

Are your expectations on oversight with regards, Project Reporting, Technology reports, Quality Assurance, understood and suitably matched with your needs? (E.g. regulatory requirements, design authority etc)

Requirements

Is a detailed statement of requirements included as part of the contract?

Does the contract provide for changes in requirements? How is this handled? Is there a formal Change Request process (Technical, Programme, Contractual)?

Will you need to provide proprietary tools for the vendor’s use?

Are your rights to those tools protected?

Deliverables

What work products aside from the software itself will be produced as part of the contract—architecture description, design description, software models, source code, inline documentation, external documentation, test plan, test cases, test results, quality metrics, user documentation, maintenance plan?

Are there test rigs, test harnesses, used by the supplier and who retains ownership/who pays for them?

Does the supplier expect to have test harnesses, test rigs or other equipment supplied from you to complete his work?

Acceptance

Is there a clearly refined set of acceptance criteria for delivery of the final product?

Are measurable quality criteria specified e.g. defect rates per LOC or MTTR?

Who is responsible for determining acceptability?

What mechanisms protect both parties from disagreements regarding the delivered product’s acceptability?

Warranty / Defect Resolution / Maintenance

Is there a warranty declaration provided by the supplier regarding the software and its usage?

If the vendor is responsible for correcting defects post delivery, does the contract specify critical repair scenarios and reaction times?

Is there a Service Level Agreement (SLA) as part of the contract or negotiated separately?

Who is responsible for maintaining the code and documentation, both during development and post installation?

Is a maintenance contract defined? If so is it on a fixed price, annual basis or is it time and materials at agreed rates?

Licensing

If the software is licensed, how are licensing fees handled?

How much, if any, will the vendor be allowed to increase licensing fees for future versions of the product?

Does the supplier have a separate Software License agreement that is in addition to or takes precedence over any development contract?

Are export controls laws covered by the contract? Is the supplier responsible for getting export license for software if required?

Ownership / IP

Who retains ownership of, or has rights to, original code written for the project?

Who retains ownership of, or has rights to, code the vendor provides from its code library? (May vary if non-exclusive e.g. if reuse of existing code base)

Will the vendor provide documentation for code that it provides from its code library?

Does the vendor have the right to sell the software developed for you to other customers? (Non-exclusivity).

Is the vendor prevented from developing similar products for your competitors? (ever, or for a fixed term of market advantage, or outside your target markets).

Who has rights to the source code if the vendor becomes insolvent? Can the source code be placed in escrow so that it can be retrieved it if that happens?

If ESCROW is mandated, is the development environment included?

Branding

Does the vendor require that their brand is visible to the end customer either via operation of the software (e.g. included in the HMI) or in product documentation?

Have the implications of branding been fully considered at a business level including the consequences on your brand image with end customer?

Employee Solicitation/Hiring

Is the vendor prevented from hiring people from you? Is the agreement reciprocal?

Additional points for shared/joint development projects

Who retains ownership of or has rights to code provided from your code library?

If you provide source code to the vendor for inclusion in the product, is the vendor restricted from reusing that source code in other products or selling that source code to other customers?

Are your developers who interact with the vendor required to sign non-disclosure agreements?

What are the implications of the vendor doing that?

Will it restrict your ability to develop your own products in the future?

Appendix A – RASIC

Definition plagiarised and modified from Wikipedia.

Responsible

Those who are responsible for the task, ensuring that it is done as per the Approver. There must be only one Responsible specified for each task or deliverable. If other parties are involved in delivering into the task then they are Support.

Accountable (also Approver or final Approving authority)

Those who are ultimately accountable for the correct and thorough completion of the deliverable or task, and the one to whom Responsible is accountable. In other words, an Accountable must sign off (Approve) on work that Responsible provides. There must be only one Accountable specified for each task or deliverable.

Support

Resources allocated to Responsible. Unlike Consulted, who may provide input to the task, Support will assist in completing the task.

Informed

Those who are kept up-to-date on progress, often only on completion of the task or deliverable; and with whom there is just one-way communication.

Consulted

Those whose opinions are sought; and with whom there is two-way communication.

Appendix B – Statement of Work

1. Preamble
2. Project Background
3. Scope
4. Key Tasks and Milestones
5. Project Deliverables
6. Time and Cost Estimates
7. Price and Payment
    a. Invoices
    b. Payment
8. Project Organization and Personnel Requirements
9. Supporting Documentation
10. Expenses

Appendix C – Supplier Capability

These questions are based on the Dominant Cost, Schedule & Risk Factors from Software cost modelling, and should be used to judge capability of competing suppliers. Absolute (rather than relative) judgement will need significant objective and subjective assessment by appropriate professionals.

The supplier shall provide the above evidence for the following risk "factors"

1. Process Capability: What is the capability of the processes to be used. What is the expected impact of improvements?

2. Technology Maturity: How tried-and-tested is the proposed solution. What is the level of technical novelty, invocation and risk?

3. Project Management Capability: What is the capability and effectiveness of the project management team, tools, methods, measures and reporting?

4. Organisational Capability: What is the maturity of the organisation intra-company-structure, off load suppliers, sub-tier suppliers i.e. are they stable, established and performance understood.

5. Customer Capability. How mature is the customer interface, key touch-points, methods for working together etc. What is the experience and capability of the customer?

6. Product Complexity: How complex is the proposed solution, number of interfaces and dependencies, logical and functional complexity etc?

7. Product volatility: what is the level of change expected during the course of the project?

8. Requirements Volatility. How mature are the customers and suppliers requirements and what is the potential for change.

9. Team Stability, Capability & Experience: The stability, competence, capability and experience of the team as a whole and individual groups e.g. key architects and designers.

10. Product Quality:: What is the expected quality of the delivered products, the number of error escapes and its acceptability to the customer, the ability to accommodate corrections, the expected level of Scrap & Rework etc

Appendix D – Intellectual property law

The four key types of intellectual property law that can affect software are as follows:

Copyrights

Software copyright is an established form of intellectual property protection. However proving infringement of software copyright is notoriously difficult and there is a mixed case history of actions against software copyright infringement e.g. http://www.patentarcade.com/2008/07/case-sega-v-accolade-9th-cir-1992.html

Patents (definition from Wikipedia)

Software patent does not have a universally accepted definition. One definition suggested by the Foundation for a Free Information Infrastructure is that a software patent is a "patent on any performance of a computer realised by means of a computer program".

There is intense debate over the extent to which software patents should be granted, if at all. Important issues concerning software patents include:

- Where the boundary between patentable and non-patentable software should lie;
- Whether the inventive step and non-obviousness requirement is applied too loosely to software; and
- Whether patents covering software discourage, rather than encourage, innovation.

Trade secret (definition from Wikipedia)

The precise language by which a trade secret is defined varies by jurisdiction (as do the particular types of information that are subject to trade secret protection). However, there are three factors that, although subject to differing interpretations, are common to all such definitions: a trade secret is information that:

- is not generally known to the public;
- confers some sort of economic benefit on its holder (where this benefit must derive specifically from its not being generally known, not just from the value of the information itself);
- is the subject of reasonable efforts to maintain its secrecy.

Trademarks (definition from Wikipedia)

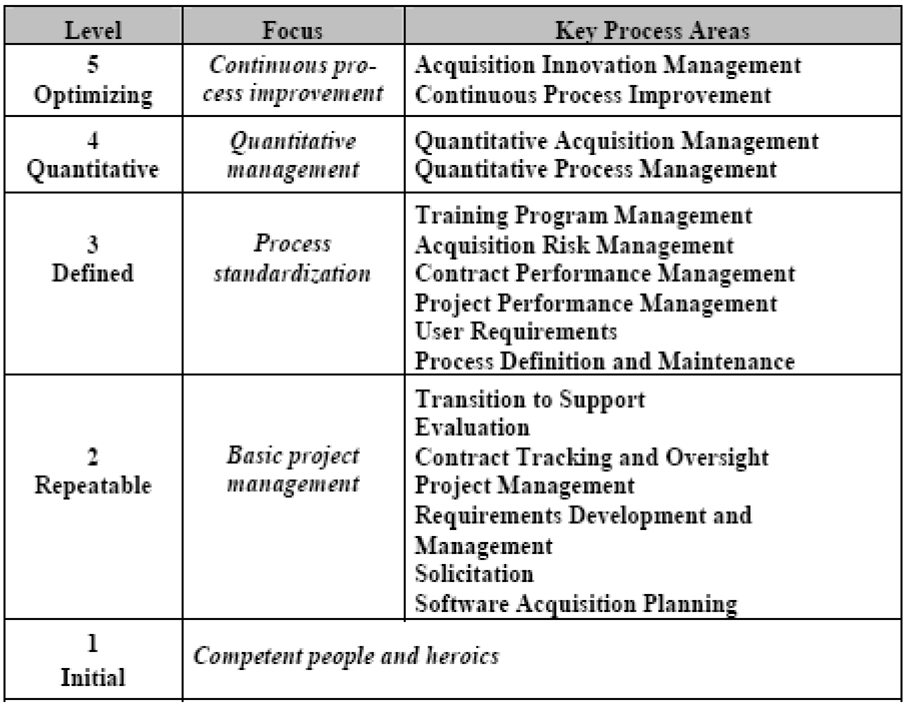
A trademark or trade mark is a distinctive sign or indicator used by an individual, business organisation, or other legal entity to identify that the products or services to consumers with which the trademark appears originate from a unique source, and to distinguish its products or services from those of other entities

Appendix E – Software Acquisition - Capability Maturity Model

The Software Engineering Institute's SA-CMM is a model for benchmarking and improving the software acquisition process.

The model follows the same architecture as the Capability Maturity Model for Software (SW-CMM), but with a unique emphasis on acquisition issues and the needs of individuals and groups who are planning and managing software acquisition efforts.

There are 5 levels of maturity defined by the model as follows:



Level	Focus	Key Process Areas
5 Optimizing	*Continuous process improvement*	Acquisition Innovation Management Continuous Process Improvement
4 Quantitative	*Quantitative management*	Quantitative Acquisition Management Quantitative Process Management
3 Defined	*Process standardization*	Training Program Management Acquisition Risk Management Contract Performance Management Project Performance Management User Requirements Process Definition and Maintenance
2 Repeatable	*Basic project management*	Transition to Support Evaluation Contract Tracking and Oversight Project Management Requirements Development and Management Solicitation Software Acquisition Planning
1 Initial	*Competent people and heroics*	

Bibliography

Software Acquisition Capability Maturity Model – SEI
http://www.sei.cmu.edu/reports/02tr010.pdf