# Integrating ISO 17799 into your Software Development Lifecycle

By Ismael Valenzuela

**It is a well-known fact in computer security that security problems are very often a direct result of software bugs. That leads security researches to pay lots of attention to software engineering. The hope is to avoid the ever present penetrate-and-patch approach to security by developing more secure code in the first place. - McGraw and Felton, 1999.**

It's no wonder that including security early in the development process will usually result in less expensive, less complex and more effective security than adding it during the lifecycle.

Given that ISO 17799 is the international code of reference for information security, we will focus on how to integrate key controls selected from such standard into all phases of the SDLC process, from initiation to disposal.

SDLC is a framework for developing software successfully that has evolved with methodologies over time. Discussions over different models are out of the scope of this article but regardless of which software development model is used, there are typical phases that need to be included. The basic phases are:

- Project initiation and functional requirements definition
- System design specifications
- Build (develop) and document
- Acceptance
- Transition to production (installation)
- Operations and maintenance support (post-installation)
- System replacement (disposal).

Therefore, to successfully include security into the SDLC process, the following requirements must be met:

- It must be based on security principles adhering to a recognized standard and information privacy.
- It must be focused on risk and compliance
- It must include activities designed to ensure compliance to ISO 17799:2005
- It must require security-related steps in SDLC procedures
- It must be supported by management as well by information and business process owners.

Before going further, we can think in ISO 17799 like ‘the security control supermarket’; hence, you go there and pick what you fancy! However, this is a peculiar supermarket in the way that you need to justify all the decisions you make on what controls you choose and what controls you don’t. This justification will be based in periodic risk assessments. This is what we call a risk based approach.

So based on these typical phases, the next section provides a correlation of where security tasks and ISO 17799 based controls should be included during the activities completed at each of these SDLC basic phases.

## Security Activities within the SDLC

### Project initiation and functional requirements definition

At the beginning phase, business needs are identified along with the proposed technical solution. The identified solution must be aligned to business strategy as well as to IT strategy and security strategy.

At this stage, ISO 17799:2005 section 6 (Organization of Information Security) highlights the importance of a well established security management framework where security related decisions are supported by the business, responsibilities are clearly allocated and activities coordinated across the organization. During this early phase of development, the organization will determine its information security requirements, often developed by successive refinement, starting from a high level of abstraction that may include the information security policy and the enterprise architecture, and then adding additional specifications during consecutive phases.

However, the definition of the security requirements must always include security categorization and a preliminary risk assessment.

According to section 7 (Asset management and information classification), to ensure that the information handled by your application receives the appropriate level of protection, you will have to identify information assets and categorize each according to regulatory impact, business criticality and sensitivity. Most of information classification schemes define three levels (low, moderate or high) of potential impact for organizations or individuals should there be a breach of security (a loss of confidentiality, integrity or availability). This standard will assist you in making the appropriate selection of security controls for your application.

On the other hand, a preliminary risk assessment will result in a brief initial description of the basic security needs of the system, expressed again in terms of the need for integrity, availability and confidentiality. This assessment will establish the threat environment in which the application will operate, followed by an initial identification of required security controls that must be met to protect it in the intended operational environment. The technical, operational and economical feasibility of these controls and any other security alternatives must be analyzed at this point. A cost / benefit analysis should be undertaken for each control, resulting in a preliminary risk treatment plan. To assess the effectiveness of those controls, a security test and evaluation plan will be developed in order to provide important feedback to the application developers and integrators at later stages.

This risk-based approach, as stated on section 4 from the standard, is the basis for any successful security initiative; hence risk assessments must be repeated periodically during the application lifecycle to address any changes that might influence the risk assessment results, until consistency is achieved.

Although you can choose among many different risk analysis methods, the risk assessment will not necessarily be a large and complex document. It is extremely easy to get lost in a complex risk analysis, so bear in mind that the risk assessment is a mean to achieve your goal but not the goal itself, so keep it simple.

In addition, the application context should be considered, as it might affect other applications or systems to which it will be directly or indirectly connected. If the context is not considered, there is a possibility that the application being developed could compromise other organization systems.

Additionally, the security functional requirements analysis should include not only a security policy and enterprise architecture analysis, but also an analysis of applicable laws and regulations, such as the Privacy Act, HIPAA, SOX, ISO 27001, and others, which define baseline security requirements. As section 15.1 (Compliance with legal requirements) states, all relevant legal and contractual requirements as long as functional and other IT security requirements should be explicitly defined, documented, and kept up to date for each information system in the organization.

A preliminary business continuity plan focused on the required business objectives is also produced at this point, e.g. restoring of specific communication services to customers in an acceptable amount of time, etc. Producing this plan requires full involvement from application and business processes owners, and again, is based on a business continuity risk assessment. A list of items that must be considered within the business continuity planning process is included in section 14.1.3 (Developing and implementing continuity plans including information security) of the standard.

Typically, a service level agreement (SLA) is also required to define the technical support or business parameters that an application service provider will provide to its clients, as well as the measures for performance and any consequences for failure. These kinds of agreements together with a typical list of terms are covered on section 6.2.3 (Addressing security in third party agreements) of the standard.

This section finalizes with the security framework documentation, resulting in a high-level description of the security issues, risks and controls in the proposed application and the assurance requirements. This material will be used to support the derivation of a cost estimate that addresses the entire life-cycle. It is usually the case that there is a balance such that increased expenditures during early development stages may result in savings during application operation.

**PRIOR THE APPROVAL OF DESIGN SPECIFICATIONS, A COMPREHENSIVE SECURITY RISK ASSESSMENT SHOULD HAVE BEEN CONDUCTED**

### System design specifications

This phase includes all activities related to actually designing the application. In this phase, the application architecture, system outputs, and system interfaces are designed while data input, data flow and output requirements are established. Detailed security specifications are included into the formal baseline documentation and the security test plan will be updated, including specific procedures on how to validate system components through development and deployment stages.

Prior the approval of design specifications, a comprehensive security risk assessment should have been conducted, including those risks related to third parties that may require access to the organization's information and information processing facilities. This assessment will result in the identification of appropriate security controls that will be agreed and included into a contract or into a SLA. A list of issues that should be taken into account before granting access to any external party are listed in section 6.2.1 (Identification of risks related to external parties) of the standard.

At this point, access control rules must be defined to ensure that access to information and information processing facilities are controlled on the basis of business and security requirements previously defined. In addition, a policy should be in place to maintain the security of information that may be exchanged through the application with any external entity. Section 10.8.1 (Information exchange policies and procedures) contains a comprehensive list of security issues that should be considered when using electronic communication facilities for information exchange, i.e. using cryptographic techniques to protect the confidentiality, integrity and authenticity of information.

Capacity management is also a key area that must be formally considered when it comes to application development. At this stage, section 10.3.1 (Capacity management) states that any projections of future capacity requirements should take account of new business and system requirements and current and projected trends in the organization's information processing capabilities. This is particularly important in case your application requires any resources with long procurement lead times or high costs.

At the end of this phase, all appropriate security controls must be defined and included into the application design specifications.

### Build (develop) and document

During this phase, the source code is generated, test scenarios and test cases are developed, unit and integration testing is conducted, and the program and system are documented for maintenance and for turnover to acceptance testing and production.

At this stage, the parallel security activities must ensure that any security-related code is actually written (or procured) and evaluated, security tests are performed and that all approved security components in formal baseline are included.

Most of security controls that will be implemented during this phase are found on section 12 (Information systems acquisition, development and maintenance) of the standard, i.e.:

- Input and Output data validation to ensure that data is correct and appropriate and to prevent standard attacks including buffer overflow and code injection.
- Validation checks to detect any corruption of information through processing errors or deliberate acts.
- Cryptographic techniques to ensure authenticity and protecting message confidentiality and integrity in applications.

Additionally, section 12.4 (Security of system files) gives some guidelines on securing access to system files and program source code. Test environments are usually complicated and difficult to manage environments, so special care should be taken to avoid exposure of sensitive data within them. It is highly recommended to avoid the use of operational databases containing personal data or any other sensitive information for testing purposes, as this could result in a breach of data protection laws, and access to program source code and associated items (such as designs, specifications, verification plans and validation plans) should be strictly controlled in order to prevent the introduction of unauthorized functionality and to avoid unintentional changes. Section 12.4.3 gives particular recommendations to control access to program source libraries in order to reduce the potential for corruption of computer programs.

### Acceptance

In the acceptance phase, an independent group develops test data and tests the code to ensure that it will function within the organization's environment and that it meets all the functional and security requirements. Prior to this stage, managers should ensure that the requirements and criteria for acceptance of new applications are clearly defined, agreed, documented and tested. It is essential that an independent group test the code during all applicable stages of development to prevent a separation of duties issue, as recommended by section 10.1.3 (Segregation of duties) of the standard.

As recommended in the previous section, any test must be carried out with previously sanitized data to ensure that sensitive production data is not exposed through the test process. A list of items that should be considered prior to formal application acceptance being provided is found on section 10.3.2 (System acceptance).

Good practice, as stated in section 10.1.4 (Separation of development, test and operational facilities), includes the testing of software in an environment segregated from both the production and development environments, as this provides a means of having control over new software and allowing additional protection of operational information that is used for testing purposes. This should include patches, service packs, and other updates.

The security testing should uncover all design and implementation flaws that would allow a user to violate the software security policy and requirements, and to ensure test validity, it should be tested in an environment that simulates the intended production environment. As a result of such tests, security code may be installed and necessary modifications undertaken.

Section 12.6.1 (Control of technical vulnerabilities) provides guidance on how to perform integrated application component tests and identifies several steps that should be followed to establish an effective management process for technical vulnerabilities, including vulnerability monitoring, vulnerability risk assessment, patching, asset tracking, etc.

This will be the last chance to detect security weaknesses or vulnerabilities as, once the application security has been verified and the system has been accepted, it will moved into production.

**SPECIAL CARE SHOULD BE TAKEN WHEN TRANSFERRING A SYSTEM FROM DEVELOPMENT TO OPERATIONAL STAGE, AS SUCH CHANGES CAN IMPACT ON THE RELIABILITY OF APPLICATIONS**

## Transition to production (installation)

During this phase the new system is transitioned from the acceptance phase into the live production environment. Typical activities during this phase include training the new users according to the implementation and training schedules; implementing the system, including installation, data conversions, and, if necessary, conducting any other parallel operations. Security control settings and switches are enabled in accordance with the defined security baseline and available security implementation guidance.

Parallel security activities verify that the data conversion and data entry are controlled and only those who need to have access during this process are allowed on the system.

Also, an acceptable level of risk is determined and accepted by business managers and appropriate controls are in place to reconcile and validate the accuracy of information after it is entered into the system. It should also be tested the ability to substantiate processing.

Special care should be taken when transferring a system from development to operational stage, as such changes can impact on the reliability of applications. Therefore, section 10.1.4 (Separation of development, test and operational facilities) gives additional recommendations that should be considered, i.e. removing any development tools or system utilities from operational systems when not required, removing development and test personnel access rights to the operational system and its information, etc.

However, even though small organizations may find enforcing segregation of duties and environments difficult to achieve, the principle should be applied as far as possible and practicable. Whenever it is difficult to segregate, as recommended by section 10.1.3 (Segregation of duties), other controls such as monitoring of activities, audit trails and management supervision should be considered.

Finally, transition to production must be controlled by the use of formal change control procedures to minimize the corruption of information systems. You will find a comprehensive list of items that the change procedures should include on section 12.5.1 (Change control procedures).

## Operations and maintenance support (post-installation)

During this phase, the application will be in general use throughout the organization. The activities involve monitoring the performance of the system and ensuring continuity of operations. This includes detecting defects or weaknesses, managing and preventing system problems, recovering from system problems, and implementing system changes.

It’s no wonder that inadequate control of changes is exactly the most common cause of system and security failures.

Therefore, to ensure the correct and secure operation of information processing facilities, any changes to systems and application software should be subject to strict change management control, as recommended by section 10.1.2 (Change management).

In general, changes to operational systems should only be made when there is a valid business reason to do so, and must be always preceded by an assessment of the potential impacts, including security impacts, of such changes. Hence, periodic risk analysis are required whenever significant changes occur, including a change in data sensitivity or criticality, relocation or major change to the physical environment, new equipment, new external interfaces, new operating system software (as considered on section 12.5.3 – Technical review of applications after operating system changes -), or new application software. It is recommended that a specific group or individual is given responsibility for monitoring vulnerabilities and vendors' releases of patches and fixes. Also, someone should be assigned the task of verifying compliance with applicable service level agreements according to the initial operational and security baselines.

Other parallel security activities considered during this stage include:

- System monitoring to detect any unauthorized information processing activities, to check the effectiveness of controls adopted, and to verify conformity to an access policy model, as stated on section 10.10 (Monitoring). This section also addresses audit logging activities, the protection of log information and logging facilities and supporting activities like clock synchronization.
- Network security management to ensure the protection of the application information and the protection of the supporting infrastructure as stated on section 10.6 (Network security management).
- Technical compliance checking to ensure that hardware and software controls have been correctly implemented and usually involving penetration tests or vulnerability assessments as considered on section 15.2.2 (Technical compliance checking). Special care should be taken when auditing to minimize the risk of disruption to business processes and access to audit tools should be protected to prevent any possible misuse or compromise. Further information on this topic can be found on section 15.3 (Information systems audit considerations).
- Documenting operating procedures as considered on section 10.1.1 (Documented operating procedures), to ensure the correct and secure operation of information processing facilities. This control also helps to ensure that system activities associated with the application (backups, maintenance, media handling, etc…) are always made available and updated to all users who need them.
- Control of operational software to minimize the risk of corruption to operational systems by implementing a rollback strategy, activating auditing logs, archiving old versions of software and other guidelines found on section 12.4.1 (Control of operational software).
- Protection against malicious and mobile code to protect the integrity of software and applications. Section 10.4 (Protection against malicious and mobile code) provides guidance on the detection, prevention and recovery controls that should be implemented across the organization.

Nevertheless, when it comes to access control, the standard allocates a whole section to this pillar of security. Controlling the allocation of access rights to information systems, privilege management, password management and the review user access rights are a day-to-day challenge. You'll find particularly useful to follow the guidelines found on section 11 (Access control) and to implement those specific controls that will be selected as a result of the previous risk assessments.

### System replacement - disposal

Disposition, the final phase in the SDLC, provides for disposal of the application in place. Information security issues associated with disposal should be addressed explicitly. In general, when information systems are transferred, obsolete, or no longer usable, it is important to ensure that organization resources and assets are protected. Generally, an application owner should archive critical information, sanitize the media that stored the information and then dispose of the hardware/software.

ISO 17799:2005 gives particular emphasis to secure disposal or re-use of equipment (section 9.2.6) when it recommends that all devices containing sensitive data should be physically destroyed or the information should be destroyed, deleted or overwritten using techniques to make the original information non-retrievable rather than using the standard delete or format function.

Finally, as stated on section 9.2.7 (Removal of property) you must remember that any equipment, storage media, information or software should not be taken off-site without prior authorization and, where necessary and appropriate, it should be recorded as being removed off-site.

| SDLC Phases | Project Activities | Parallel Security Activities | ISO 17799:2005 mapping |
|---|---|---|---|
| Project Initiation and Functional Requirements Definition | • Identify business needs<br>• Identify areas affected and responsibilities<br>• Develop functional requirements<br>• Propose technical solution<br>• Evaluate alternatives<br>• Document project's objectives, scope, strategies, costs and schedule.<br>• Select / approve approach<br>• Prepare project plan<br>• Prepare preliminary test plan<br>• Select acquisition strategy<br>• Establish formal functional baseline | • Determine security requirements<br>• Classification and criticality of information/applications<br>• Identify legal, statutory and contractual requirements<br>• Initial Risk Analysis<br>Cost / benefit analysis<br>• Preliminary contingency planning<br>• Prepare a security evaluation plan<br>• Include security requirements in the security baseline as well as in request for proposal and contracts<br>• Determine SLAs<br>• Document security framework | 5.1.1 – Security Policy<br>7.x – Asset management and information classification<br>6.1.1, 6.1.2, 6.1.3 – Organization of Information Security<br>12.1 – Security requirements of information systems<br>6.2.3 – Addressing security in third party agreements<br>15.1 – Compliance with legal requirements<br>14.1.3 – Business Continuity Management |
| System Design Specifications | • Develop detailed design (system architecture, system outputs and system interfaces).<br>• Detail the solution's interactions with external systems.<br>• Update testing goals and plans. Establish data input, data flow and output requirements.<br>• Establish formal baseline/ quality controls and requirements. | • Identification of Risks related to external parties.<br>• Define access control strategy<br>• Define security specifications (program, database, hardware, firmware and network)<br>• Develop security test procedure<br>• Include security area in formal baseline documentation and quality assurances | 11.1 – Business requirement for access Control<br>6.2.1 – Identification of risks related to external parties<br>10.8.1 – Information exchange policies and procedures<br>10.3.1 – Capacity management |
| Build/Development and Documentation | • Construct source code from detailed design specifications.<br>• Perform and evaluate unit tests.<br>• Implement detailed design into final system. | • Write or procure and install security-related code.<br>• Control access to code.<br>• Evaluate security-related code.<br>• Ensure approved security components in formal baseline are included. | 12.2.x –Correct processing in Applications<br>12.3.x – Cryptographic controls<br>12.4.x – Security of System Files |

| SDLC Phases | Project Activities | Parallel Security Activities | ISO 17799:2005 mapping |
|---|---|---|---|
| Acceptance | • Test system components.<br>• Validate system performance.<br>• Install system.<br>• Prepare project manuals.<br>• Perform acceptance  test.<br>• Accept system. | • Sanitize test data.<br>• Independent security tests.<br>• Install security code with necessary modifications.<br>• Document security controls. | 10.3.2 – System acceptance<br>12.6.1 – Technical vulnerability management<br>10.1.4 – Separation of development, test and operational facilities |
| Transition to Production (implementation) | • Train new users according to implementation.<br>• Implement the system (installation, data conversions...). | • Control data conversion and data entry.<br>• Reconcile and validate data integrity.<br>• Enforce segregation of duties and segregation of environments. | 12.5.1 – Change control procedures<br>10.1.3 – Segregation of duties<br>10.1.4 – Separation of development, test and operational facilities |
| Operations and Maintenance Support (post-installation) | • Monitoring performance.<br>• Ensuring continuity of operations.<br>• Detect weaknesses or defects.<br>• Manage and prevent system problems.<br>• Recover from system problems.<br>• Implement system changes. | • Periodic risk analysis.<br>• Change management.<br>• Verify compliance with applicable SLAs and security baselines.<br>• Maintain release integrity with secure and controlled environments. | 10.10. x – Monitoring<br>12.5.2 – Technical review of applications after operating system changes<br>10.6.x – Network security management<br>11.x – Access control<br>15.2.2 – Technical compliance checking<br>15.3.x – Information systems audit considerations<br>10.1.1 – Documented operating procedures<br>12.4.1 – Control of operational software<br>10.4.x  - Protection against malicious and mobile code |
| System Replacement - Disposal | • Hardware and Software disposal. | • Information preservation.<br>• Media sanitization. | 9.2.6 – Secure disposal or re-use of equipment<br>9.2.7 – Removal of property |

## Conclusion

Introducing a risk management program along all your project phases is the key to success in introducing security into SDLC. However, we must admit that it can be challenging on an uncompleted cycle and identification of mitigation points is sometimes tricky. Additionally, adding an IT process-centered practice approach, like ITIL and COBIT, aids in being able to determine how best to embed security controls into your operational processes and how to measure their effectiveness. Security awareness is another driving factor and collaboration between all parties, the business and ICT department, is critical.  Management must be clearly committed to information security and managers must be made responsible and accountable for the security of their application systems. They should ensure that all proposed system changes are reviewed to check that they do not compromise the security of either the system or the operating environment.

Ismael is working as an Information Security Specialist at iSOFT plc. He is involved in security governance and compliance, implementing ISO 27001, providing risk assessment and security consulting and strategy. He has also participated as a senior consultant for many big security projects in Spain, as well as an instructor, writing articles and promoting security business. Ismael holds a Bachelor in Computer Science, is certified in Business Administration, ITIL, CISM and CISSP and was recently accredited as IRCA ISO 27001 Lead Auditor by Bureau Veritas UK. Ismael can be reached at ismael.valenzuela@gmail.com and www.linkedin.com/in/ivalenzuela.