SSA Integration in Delivery Lifecycle

SSA helps to ensure that the required security controls are built into the design as per security principles, implemented as per secure programming practices, and verified against customer requirements and industry best practices. Initial version of SSA was influenced by Microsoft SDL 3.2.

Contract / Proposal		Education	Requirement	Design	Development	Testing	Deployment	Maintenance
• Classify software based on risk (criticality) • Estimate security effort	Low Risk	• Weakness awareness • Security testing • Secure programming & code review	• Security requirements elicitation	• Secure design	• Secure programming • Compilation with security options	• Automated security testing	• Infrastructure VA (Automated)	• System health report • Access analysis • Incident response • Log analysis *
• Signoff from GL on security effort estimation	Medium Risk	• Security engineering • Software security risk management	• Analysis using formal method such as abuse cases *	• Review using formal method such as threat modeling *	• Static code analysis • Security code walkthrough	• Manual security testing	• Infrastructure VA (Manual)	• Change effectiveness analysis
• Signoff from ISM on compliance requirements	High Risk	• Security architecture design • Compliance	• Signoff from customer security team	• Security architecture & API design • Design review by external team	• Security code walkthrough by external team	• Security testing by external team	• Infrastructure VA by external team	• Resource usage analysis
• Signoff from OU risk officer	Very High Risk	• Security workshop to explain each mission critical requirement in detail with security expert	• Signoff from customer legal team (if applicable)	• Design review workshop and signoff from customer security team	• Line by line analysis of source code & configuration	• Penetration testing by third party	• Penetration testing by third party	• Periodic security workshop with customer

Contract / Proposal		Education	Requirement	Design	Development	Testing	Deployment	Maintenance
• Classify software based on risk (criticality) • Estimate security effort	Low Risk	• Weakness awareness • Security testing • Secure programming & code review	• Security requirements elicitation	• Secure design	• Secure programming • Compilation with security options	• Automated security testing	• Infrastructure VA (Automated)	• System health report • Access analysis • Incident response • Log analysis *
• Signoff from GL on security effort estimation	Medium Risk	• Security engineering • Software security risk management	• Analysis using formal method such as abuse cases *	• Review using formal method such as threat modeling *	• Static code analysis • Security code walkthrough	• Manual security testing	• Infrastructure VA (Manual)	• Change effectiveness analysis
• Signoff from ISM on compliance requirements	High Risk	• Security architecture design • Compliance	• Signoff from customer security team	• Security architecture & API design • Design review by external team	• Security code walkthrough by external team	• Security testing by external team	• Infrastructure VA by external team	• Resource usage analysis
• Signoff from OU risk officer	Very High Risk	• Security workshop to explain each mission critical requirement in detail with security expert	• Signoff from customer legal team (if applicable)	• Design review workshop and signoff from customer security team	• Line by line analysis of source code & configuration	• Penetration testing by third party	• Penetration testing by third party	• Periodic security workshop with customer

SSA is an integral part of TCS iQMS (from 2011) and is revised on a quarterly basis to protect TCS deliveries against emerging threats. The SSA Process Handbook in iQMS Wiki describes activities in the ETVX format. Following is a brief description of the SSA Process Handbook

Software Security Integration

This phase ensures that the security principles are adhered to, at every step within the development life cycle. It enables the user to identify and document SSA deliverables, milestones and activities to ensure that security is well integrated with Software Development Life Cycle (SDLC). It serves as an overall guidance to the project team by defining security objectives, effort estimation, evaluation criteria for each deliverable, budgeting, security review schedules and involvement of external teams. It also ensures that the customer security expectations are neither overlooked nor underestimated from the management perspective.

Following is the scope of this phase:

- Including security assurance into project plan and effort estimation
- Setting up security lab for carrying out security assurance tasks
- Facilitating security mentorship and identifying security training needs
- Identifying the need for independent review/third party review

Security Requirements Analysis

This phase ensures that security, privacy, reliability and compliance related requirements are well understood, documented, agreed, signed off, and provided as input to the Architecture and Design team. It helps users to understand the security and compliance requirements like security features and functions, privacy requirements, banned APIs/components/libraries, security constraints identified during functional / non-functional requirement elicitation, impact of security on other quality attributes, precaution suggested after risk assessment, security best practices and special training needs, development and review methodology, architectural limitations, deployment strategy, security acceptance criteria and involvement of external teams. The project team identifies security requirements through various techniques such as customer interview, policy/standard compliance, use/abuse cases, and brainstorming sessions.

Following is the scope of this phase:

- Including security, privacy, reliability and compliance requirements into specification.
- Developing security requirements from function requirements using formal methods such as abuse case analysis.
- Identifying constraints, limitations and trade-offs for security requirements.
- Developing delivery acceptance criteria.

Architecture / Design Security Analysis and Review

This phase identifies threats and design flaws. It provides pragmatic inputs for improving architecture/design while maintaining its focus on security requirements and minimum conflict with other attributes like quality, performance, and so on. It helps user to analyse proposed architecture / design to identify flaws and potential threats, recommend pragmatic mitigation for improvement or reengineering while keeping focus on security requirements with minimum conflict on other quality attributes.

Following is the scope of this phase:

- Reviewing application environment, architecture and design.
- Applying techniques such as attack surface identification, trust validation, data flow analysis, dependency analysis for interfaces, security controls, privileges, PII and business critical data, external code/library integration, data disclosures, message integrity, storage, audit trail and configuration.
- Analysing using formal methods such as threat modeling to ensure that potential threats and associated risks are identified and mitigation security controls considered to bring risk at acceptable level.

Secure Code Construction and Security Code Review

This phase enforces secure coding, reduces implementation-level vulnerabilities and recommends remedies to the Development team for rectifying coding errors. It helps writing secure code and reviewing the source code to uncover typical mistakes and common erroneous practices followed during construction. It verifies that the proper security controls are present, work as intended and are invoked by authorised routines in the given context. It also helps to identify malicious code (such as logic bomb). Security code review is performed combining automated static analysis for language specific vulnerable constructs and security code walk through for logically faulty constructs based on environment and application context.

Following is the scope of this phase:

- Ensuring defensive programming and compilation using security switches

- Securing configurations and dependencies
- Facilitating automated static code analysis and manual security code review

Security Testing

This phase determines whether the software is secure enough to be released or deployed. It helps in verifying the robustness of the design, accuracy and completeness of code to ensure that software protects data and maintains functionality as intended.

Following is the scope of this phase:

- Providing test plan and test cases for security function testing.
- Facilitating fuzz testing, regression testing, privacy and data security testing, application penetration testing, logical layer testing, network traffic manipulation and analysis.
- Reviewing software documentation for adequacy and accuracy for security.

Deployment / Release Security Review

This phase ensures that software deployment or installation does not weaken the target environment. It helps in verifying that the software deployment, configuration or installation does not weaken the target environment. Software binary is verified for virus/spyware, integrity, unauthorised modification on target system and change in security settings of target environment.

Following is the scope of this phase:

- Confirming known alteration on target system and running a VA/PT.
- Verifying Binary integrity
- Verifying that the software does not erase / overwrite any data / files without explicit approval from target system administrator
- Reviewing by privacy team when any private or sensitive data is collected either during installation or execution, or software deals with PII.

Maintenance (Operational) Security Review

This phase ensures that the software continues to operate in a secure environment where resources are managed efficiently, intrusions are identified before damage is caused, and incidents are handled rigorously. Despite all efforts to ensure security, vulnerabilities are discovered and new attack techniques derived over a period of time. Operational systems security helps in maintaining the softwares robustness with changes in operational environment. Operating environment includes operating systems, runtime environment, application, dependencies such as application servers, middleware platforms, cloud services, databases, COTS and so on.

Following is the scope of this phase:

- Ensuring day-to-day operations like network and operating system security, security for operational software, production code and application change management, production data usage controls, system availability and production error message sanitization for development team.
- Ensuring configuration security and log analysis, periodic activities including assessment and audit of technical as well as administrative controls of production systems, and change impact analysis for system enhancements
- Providing access reconciliation and analysing resource usage.
- Ensuring software updates, applying/releasing patches for vulnerability and retiring old version of software.
- Providing incident response, forensics, business continuity planning and mock drills to ensure that the operational environment is ready to face challenges and meet user expectations.

Security in Acquiring Software/Code for Maintenance Service

This phase ensures that security risk associated with software/code acquired for maintenance service is known. As part of software maintenance service, software and code is handed over to delivery teams and expected

to be operated securely. Software portfolio generally contains diverse set of technologies and blend of legacy, COTS/MOTS/FOSS, custom developed software and software services. It is important to gauge security posture before acquiring such software.

Following is the scope of this phase covers:

- Reviewing existing security practices and identifying improvements
- Analysing application portfolio for known vulnerabilities
- Assessing software for finding software vulnerabilities
- Estimating effort for remediating vulnerabilities
- Budgeting for security tools or services
- Estimating compensatory controls

Security in Software Retirement

This phase ensures that the software, data and dependencies are protected from unauthorised access during inactive state. Legacy software is often maintained in inactive state to enable access to data which is not migrated to newer systems. These could include data retained for legal/regulatory or analysis/business purposes and maintaining security for such system often increases operational overheads and requires additional budget. Software retirement (also called archival or decommissioning) is a process to shutdown redundant or obsolete software while protecting access to data.

Following is the scope of this phase:

- Archiving binaries, data, configurations, dependencies, tools, runtime environments, keys, installation and usage documentation / training manuals, known issues and remediation documentation, special hardware (if required), and so on.
- Transferring data to new data repository or archive store so that it can be accessed through standard reporting tools.
- Discussing/negotiating with vendors of dependent products/runtime. Unlike hardware, software is a licensed asset and most of the time license is given as right to use the software for specific period.
- Restoring to ensure integrity, understand challenges during the process of rebuilding the software and data retrieval, gauging restoration duration and skillset to operate the software.

Security in Software Disposal

This phase ensures that software and data are destroyed permanently and cannot be recovered even through forensics means. It is a process to discard system information and software. This process is generally initiated after successful execution of software retirement or new system setup.

Following is the scope of this phase:

- Destructing software information including but not limited to user data, configurations, software keys, user and software logs, software runtime setting and so on.
- Destructing software binary, documentation and any software specific references.
- Destructing licensed software (in consultation with vendor when applicable).
- Destructing special hardware such as tokens, readers and so on.

Security in COTS/MOTS/FOSS Software Acquisition

This phase ensures that the acquired software meets organisational security objectives. It is a process for acquiring Commercial-of-the-shelf (COTS), Modified off-the-shelf (MOTS) or Freeware/Open Source Software (FOSS). Software produced with security as focus reduces overall cost of ownership in long run.

Following is the scope of this phase:

- Ensuring availability of built-in defences such as IP based restrictions, certificate based access, Authentication, Authorisation, Auditing, Encryption and so on.
- Ensuring third party certification or endorsing from reputed security group about security of software.
- Ensuring software development process and security inclusion in delivery lifecycle.
- Ensuring Software release process and confirmation about no malicious code is embedded.
- Ensuring historical vulnerability check and commitment for closing security vulnerabilities with reasonable SLAs.
- Ensuring readiness of security documentation and hardening checklist.
- Ensuring availability of operational training with security.

Security in Software Service Acquisition

This phase ensures that the acquired software service meets organisational security objectives. It is a process to acquire external services for business purpose. Unlike home grown application, acquired software services may not provide organisations with detailed insight into the infrastructure, functional availability and data protection.

Following is the scope of this phase:

- Include scope from security of COTS/MOTS/FOSS Software Acquisition
- Reviewing customer data protection policies and controls based on geographical and country specific laws, contractual obligations, prohibition against cross border transfers, industry standards and so on.
- Ensuring controls for protecting confidentiality, integrity, availability and accountability
- Ensuring periodic review and analysis of security audit reports provided by service provider (as applicable).
- Ensuring periodic review and analysis of change reports provided by service provider (as applicable)
- Providing support during security incident investigations.
- Ensuring SLAs for software service availability.