Project no: 269317

nSHIELD

new embedded Systems arcHItecturE for multi-Layer Dependable solutions

Instrument type: Collaborative Project, JTI-CP-ARTEMIS

Priority name: Embedded Systems

D6.1: Lifecycle and SPD Support Plan

Due date of deliverable: M18 – 2013.02.28

Actual submission date: M19 – 2013.03.29

Start date of project: 01/09/2011 Duration: 36 months

Organisation name of lead contractor for this deliverable:

Fundación Tecnalia Research & Innovation, TECNALIA

Revision [Final 1.0]

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2012)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	X

Document Authors and Approvals

Authors		Date	Signature
Name	Company		
Arkaitz Gamino	Tecnalia	09/11/12	
Inaki Eguia	Tecnalia	09/11/12	
Balázs Berkes	S-LAB	20/02/13	
Nikos Pappas	HAI	27/03/13	
Reviewed by			
Name	Company		
Iñaki Egui	Tecnalia		
Balázs Berkes	S-LAB		
Approved by			
Name	Company		

Applicable Documents

ID	Document	Description
[01]	TA	nSHIELD Technical Annex
[D2.1]	Preliminary System Requirements	nSHIELD Preliminary System Requirements
[D2.3]	Preliminary System Architecture Design	nSHIELD Preliminary System Architecture that should be linked to SPD metrics
[D.2.2.1]	SPD Metrics for pSHIELD	SPD metrics identification for pSHIELD project
[D2.2]	Preliminary System Requirements and Specifications	nSHIELD preliminary specifications
[D2.4]	Reference system architecture design	nSHIELD document related to main outcome: reference architecture
ISO12207-2008	Systems and Software lifecycle engineering	ISO for specifying lifecycle of systems. ISO from which we can extrapolate towards SPD built-in engineering [01]
ISO 15288:2008	Systems and software engineering - System life cycle processes	It is referred to the systems life cycle process which could be indeed the reference for nSHIELD project. [02]

Modification History

Issue	Date	Description
Draft A 0.1	09.11.2012	First ToC
Draft A 0.2	15.01.2013	Contents and Plan Methodology
Review A 0.2	20.02.2013	Review by S-LAB
Review A 0.5	25.02.2013	Methodology contents by Tecnalia
Draft A 0.6	18.03.2013	Plan execution
Draft A 0.6	27.03.2013	Additions in Plan Methodology stages
Final 1.0		

Executive Summary

The purpose of the deliverable is to provide the guidelines and the plan for checking the nSHIELD SPD architecture to be future proof, and close the systems engineering life cycle by supporting the installation, downloading and upgrading cycle and addressing the security and integrity issues involved.

For the purpose of the planning for SPD Lifecycle and support, we consider two main categories of nSHIELD users: nSHIELD systems users and clients of nSHIELD systems users. Scenarios should be taken into account for this and they are key for the success of the plan developed within this document.

This plan represents an iterative process fully integrated with the development of the nSHIELD product / solution (system). In this document we analyse the methodology and the metrics to be employed in SPD Lifecycle plan. The process can be split into three subsequent phases: plan preparation, plan execution and data analysis/ assessment.

For this last phase, qualitative and quantitative evaluation criteria will be introduced.

A large part of the deliverable is devoted to the SPD lifecycle plan in the main parts of the nSHIELD system with specific focus on the subsystems realizing the interaction between the users and the system (SoS scenarios where multiple and heterogeneous devices develop a functional approach). For each of them we identified the parameters to be tested in various usage scenarios.

Contents

Figures

Tables

Glossary

Please refer to the Glossary document, which is common for all the deliverables in nSHIELD.

1 Introduction

The purpose of this document is to describe the methodology and the plan for the SPD Lifecycle and support aiming at verifying the usability and the acceptance of the solutions developed in the nSHIELD project by its main users.

This document aims to be structured through formalization; therefore the reader can understand the formal fundamentals of nSHIELD System usage. This document is about the plan to execute and implement that methodology.

It is well known that either the engineering process or about all the technology adoption is being satisfied by final users. nSHIELD System aims to unite *Security by default* focus. Embedded Systems engineering usually has been devoted to its functionality and few times has incorporated security built-in techniques. Embedded Systems are systems that were specified according to safety requirements but no security structures were taken into account for these tasks. The adoption of built-in security in Embedded Systems is the main challenge of nSHIELD that must be addressed also by the point of view of the future proof, installation, download and upgrade cycle: actually we are defining the operational view of software/hardware co-deployment approach.

1.1 Structure of this document

This document is structured as follows:

- Section 2 defines scope and context of this document
- Section 3 highlights terms and definition used in this document and in document 6.4.
- Section 4 is related to nSHIELD SPD principles
- Section 5 address the plan methodology itself
- Section 6 describes how it should be executed.

1.2 Involved stakeholders

The possible stakeholder involved during the execution of this SPD lifecycle support plan are:

Table 1-1: Roles of different stakeholders using nSHIELD system

Role	Description
Chief Information Officer (CIO)	The CIO is responsible for the organization’s information system planning, budgeting, investment, performance, and acquisition.
Configuration Management (CM) Manager	The CM manager is responsible for managing the effects of changes or differences in configurations on an information system or network.
Contracting Officer	The Contracting Officer is the person who has the authority to enter into, administer, and/or terminate contracts and make related determinations and findings.
Information System Security Officer	The Information System Security Officer is responsible for ensuring the security of an information system throughout its life cycle.
Legal Advisor/Contract Attorney	The legal advisor is responsible for advising the team on legal issues during the acquisition process.

Program Manager / Official (Information Owner)	This person represents business and programmatic interests in the information system during the SDLC process. The program manager plays an essential role in security and is, ideally, intimately aware of functional system requirements.
QA/Test Director	The QA/Test Director is responsible for system test and evaluation, and functions as a resource across a variety of programs by assisting in the development and execution of test plans in conjunction with Program Managers and customers.
Software Developer	The developer is responsible for programmatic coding regarding applications, software, and Internet/intranet sites, including “secure coding,” as well as coordinating and working with the Configuration Management (CM) manager to identify, resolve, and implement controls and other CM issues.
System Architect	As the overall designer and integrator of the application, the system architect is responsible for creating the overall design architecture and for maintaining the conceptual integrity of the architecture throughout the project life cycle.
System Owner	The system owner is responsible for the procurement, development, integration, modification, operation, and maintenance of an information system.
Other Participants	The list of SDLC roles in an information system development can grow as the complexity increases. It is vital that all development team members work together to ensure that a successful development is achieved.

1.3 References and standards

1.3.1 ISO/IEC 12207

ISO/IEC 12207 is an international standard, which defines 43 system and software processes in its derived document, the aforementioned “2008 Systems and software engineering – Software life cycle processes" [01].

This standard must be a reference model to look at in order to conclude that NSHIELD SPD terminology follows a similar criterion. The standard establishes a strong relation between software and systems. It is based upon the general principles of systems engineering. Software is treated as an integral part of the total system and performs certain functions in that system.

> This is implemented by extracting the software requirements from the system requirements and design, producing the software, and integrating it into the system. It is a fundamental premise of this standard that software always exists in the context of a system, even if the system consists of only the processor upon which the software is executed. Therefore, a software product or service is always treated as one item in a system.

It turns out that embedded systems encompass both software and hardware conditions. However, we are pointing out this document due to the **methodology for making operational nSHIELD Systems as a whole**. This is a big challenge that requires a plan to explicitly explain the main points in order to make effective the SPD functionalities within embedded systems environment.

1.3.2 ISO/IEC 15288:2008

ISO/IEC is referred to the standard for defining four categories of processes [02]:

- Technical
- Project
- Agreement
- Enterprise

The following processes are defined during the engineering lifecycle process:

Table 1-2: Processes within ISO/IEC 15288

Nº	Process name
1	Stakeholder Requirements Definition Process (Clause 6.4.1)
2	Requirements Analysis Process (Clause 6.4.2)
3	Architectural Design Process (Clause 6.4.3)
4	Implementation Process (Clause 6.4.4)
5	Integration Process (Clause 6.4.5)
6	Verification Process (Clause 6.4.6)
7	Transition Process (Clause 6.4.7)
8	Validation Process (Clause 6.4.8)
9	Operation Process (Clause 6.4.9)
10	Maintenance Process (Clause 6.4.10)
11	Disposal Process (Clause 6.4.11)

These 11 processes will be used for this document as a basis for defining SPD Lifecycle support.

1.3.3 Security Life Cycle

There are several views for Security Life Cycle definition. There are many methods and methodologies with different approaches and tackling diverse processes. The purpose of these methodologies is to add security as by default so that the process engineering can engage built-in security and not as a final patch at the last stage of the engineering process.

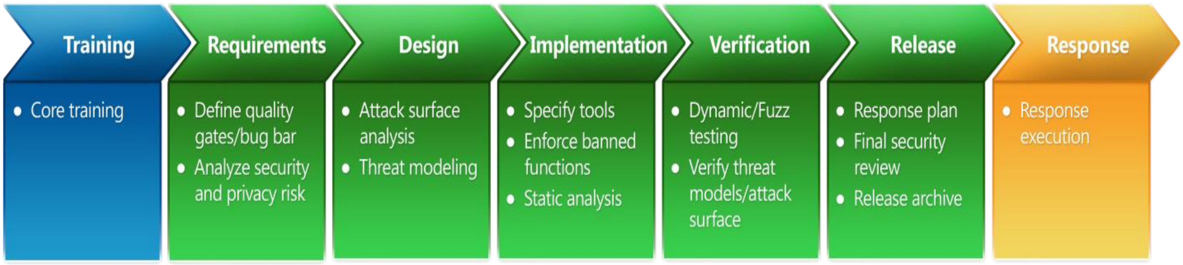

Figure 1-1: Secure software development process model at Microsoft [03]

Security methods are taken into account in order to include Security from requirements phases until deployment ones. Security needs to be addressed as a continued lifecycle to be effective. nSHIELD project is about inserting security, privacy and dependability issues into engineering process in a “systems of systems” environment. This SoS environment is an unknown and unpredictable setting that has to be addressed with novel methodologies in term of security such as, bringing to front concepts such as trustworthiness and reputation.

This new interpretation of security, privacy and dependability terms included in the lifecycle support is what this documents intends to plan.

NIST [04] includes 5 potential phases for defining security lifecycle support:

PHASE (I): INITIATION

PHASE (II): DEVELOPMENT/ACQUISITION

PHASE (III): IMPLEMENTATION / ASSESSMENT

PHASE (IV): OPERATIONS AND MAINTENANCE

PHASE (V): DISPOSAL

To the degree that we can analyse, although names might differ, they refer to similar concepts: inclusion of security from early to final stages within the engineering process.

2 Terms and definitions

Table 2-1: Terms and Definitions

Nº	Concept	Definition
1	**Acquirer**	Stakeholder that acquires or procures a product or service from a supplier
2	**Acquisition**	Process of obtaining a system, software product or software service
3	**Activity**	Set of cohesive tasks of a process
4	**Agreement**	Mutual acknowledgement of terms and conditions under which a working relationship is conducted
5	**Audit**	Independent assessment of software products and processes conducted by an authorized person in order to assess compliance with requirements
6	**Baseline**	Specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures
7	**Configuration item**	Entity within a configuration that satisfies an end use function and that can be uniquely identified at a given reference point
8	**Contract**	Binding agreement between two parties, especially enforceable by law, or a similar internal agreement wholly within an organization
9	**Customer**	Organization or person that receives a product or service
10	**Developer**	Organization that performs development tasks (including requirements analysis, design, testing through acceptance) during a life cycle process
11	**Enabling system**	System that supports a system-of-interest during its life cycle stages but does not necessarily contribute directly to its function during operation
12	**Evaluation**	Systematic determination of the extent to which an entity meets its specified criteria
13	**Facility**	Physical means or equipment for facilitating the performance of an action, e.g. buildings, instruments, tools
14	**Firmware**	Combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device
15	**Implementer**	Organization that performs implementation tasks
16	**Life cycle**	Evolution of a system, product, service, project or other human-made entity from conception through retirement
17	**Life cycle model**	Framework of processes and activities concerned with the life cycle that may be organized into stages, which also acts as a common reference for communication and understanding
18	**Maintainer**	Organization that performs maintenance activities
19	**Monitoring**	Examination of the status of the activities of a supplier and of their results by the acquirer or a third party
20	**Non-deliverable item**	Hardware or software product that is not required to be delivered under the contract but may be employed in the development of a software product
21	**Off-the-shelf**	<product> already developed and available
22	**Operator**	Entity that performs the operation of a system

23	**Organization**	Person or a group of people and facilities with an arrangement of responsibilities, authorities and relationships
24	**Party**	Organization entering into a contract
25	**Process**	Set of interrelated or interacting activities which transforms inputs into outputs
26	**Process purpose**	High level objective of performing the process and the likely outcomes of effective implementation of the process
27	**Process outcome**	Observable result of the successful achievement of the process purpose
28	**Product**	Result of a process
29	**Project**	Endeavour with defined start and finish dates undertaken to create a product or service in accordance with specified resources and requirements
30	**Project portfolio**	Collection of projects that addresses the strategic objectives of the organization
31	**Qualification**	Process of demonstrating whether an entity is capable of fulfilling specified requirements
32	**Qualification requirement**	Set of criteria or conditions that have to be met in order to qualify a software product as complying with its specifications and being ready for use in its target environment or integration with its containing system
33	**Qualification testing**	Testing, conducted by the developer and witnessed by the acquirer (as appropriate), to demonstrate that a software product meets its specifications and is ready for use in its target environment or integration with its containing system
34	**Quality assurance**	All the planned and systematic activities implemented within the quality system, and demonstrated as needed, to provide adequate confidence that an entity will fulfil requirements for quality
35	**Release**	particular version of a configuration item that is made available for a specific purpose (for example, test release)
36	**Request for proposal**	Document used by the acquirer as the means to announce its intention to potential bidders to acquire a specified system, software product or software service
37	**Resource**	Asset that is utilized or consumed during the execution of a process
38	**Retirement**	withdrawal of active support by the operation and maintenance organization, partial or total replacement by a new system, or installation of an upgraded system
39	**Security**	Protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them
40	**Service**	Performance of activities, work, or duties associated with a product
41	**Software item**	Source code, object code, control code, control data, or a collection of these items
42	**Software product**	Set of computer programs, procedures, and possibly associated documentation and data
43	**Software unit**	Separately compilable piece of code
44	**Stage**	Period within the life cycle of an entity that relates to the state of its description or realization

45	**Stakeholder**	Individual or organization having a right, share, claim or interest in a system or in its possession of characteristics that meet their needs and expectations
46	**Statement of work**	Document used by the acquirer as the means to describe and specify the tasks to be performed under the contract
47	**Supplier**	Organization or individual that enters into an agreement with the acquirer for the supply of a product or service
48	**System**	Combination of interacting elements organized to achieve one or more stated purposes
49	**System element**	Member of a set of elements that constitutes a system
50	**Task**	Requirement, recommendation, or permissible action, intended to contribute to the achievement of one or more outcomes of a process
51	**Test coverage**	Extent to which the test cases test the requirements for the system or software product
52	**Testability**	Extent to which an objective and feasible test can be designed to determine whether a requirement is met
53	**User**	Individual or group that benefits from a system during its utilization
54	**Validation**	Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled Validation in a life cycle context is the set of activities ensuring and gaining confidence that a system is able to accomplish its intended use, goals and objectives
55	**Verification**	Confirmation, through the provision of objective evidence, that specified requirements have been fulfilled Verification in a life cycle context is a set of activities that compares a product of the life cycle against the required characteristics for that product. This may include, but is not limited to, specified requirements, design description and the system itself.
56	**Version**	Identified instance of an item

3 SPD Lifecycle principles in nSHIELD

nSHIELD aims to work with the following principles following consortium criteria and taking into account Technical Annex and other systems criterions [08]:

3.1.1 Security, privacy and dependability by Design

- **nSHIELD SPD-based design and architecture.** Developers consider security issues part of the basic architectural design of software development. All architecture components are based on integrating security, privacy and dependability by design. It refers to a SPD built-in architecture.
- **Threat modelling, SPD metrics design and mitigation.** Threat models are created, and threat mitigations are present in all design and functional specifications according to specific metric defining SPD domain.
- **SPD Components design**. SPD components will be designed and will refer to SPD functionalities that might be composed (orchestrated or choreographed)
- **Improvements in security with respect legacy systems.** nSHIELD SPD node will be designed according to nSHIELD requirements but also will be able to interoperate by design with existing nodes through nSHIELD gateway.
- **Provide notice and consent.** Provide appropriate notice about data that is collected, stored, or shared so that users can make informed decisions about their personal information.

3.1.2 Security, privacy and dependability by Default

- **SPD Certification.** All components run with an accredited minimum SPD means.
- **Availability as core of SoS attributes.** nSHIELD systems must be resilient and therefore permit a fault tolerance and availability
- **nSHIELD metrics parameterizing SoSs.** The development team is aware of the attack surface for the product and minimizes it in the default configuration.
- **Avoidance of risky sub-systems incorporation and acquisition.** Reputation techniques will be used for enabling the entrance of new systems and their deployment.

3.1.3 Security, privacy and dependability in Deployment

- **Deployment guides.** Prescriptive deployment guides outline how to deploy each SoS component securely, including providing users with information that enables them to assess the security risk of activating non-default options (and thereby increasing the attack surface).
- **nSHIELD panel control and management tools.** Security analysis and management tools enable administrators to determine and configure the optimal security level for a software release.
- **Patch deployment tools.** Deployment tools are provided to aid in patch deployment. This should be a main management service that must be provided with nSHIELD solutions.

4 Plan Methodology

The methodology followed will be a conjunction of NIST and ISO provided methodologies. This will have emphasis in 5 main areas and focusing in 11 processes for all these areas. The following picture defines the methodology of NSHIELD for SPD Lifecycle Support [05], [06], [07]:

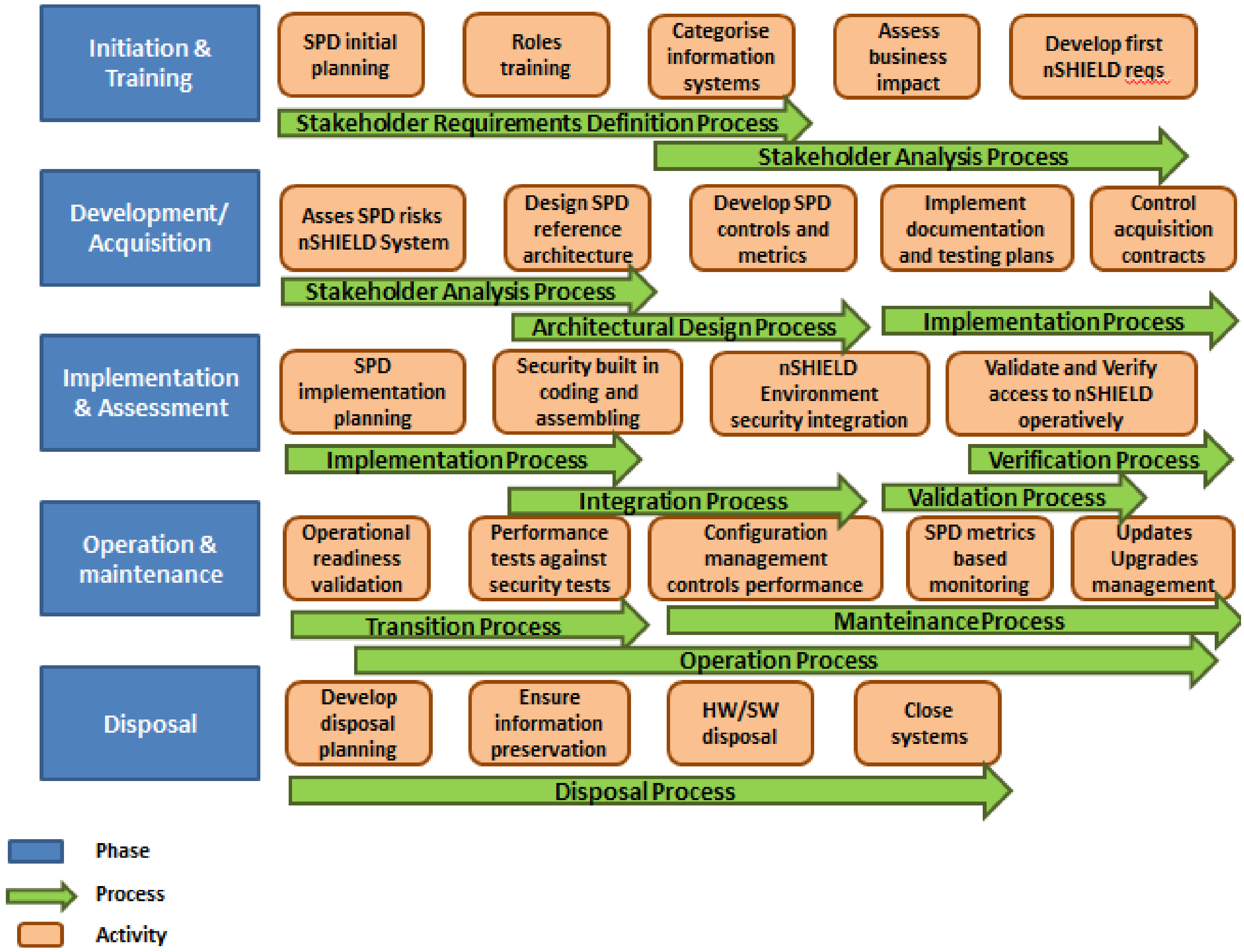
Initiation & Training
SPD initial planning
Roles training
Categorise information systems
Assess business impact
Develop first nSHIELD reqs
Stakeholder Requirements Definition Process
Stakeholder Analysis Process
Development/ Acquisition
Asses SPD risks nSHIELD System
Design SPD reference architecture
Develop SPD controls and metrics
Implement documentation and testing plans
Control acquisition contracts
Stakeholder Analysis Process
Architectural Design Process
Implementation Process
Implementation & Assessment
SPD implementation planning
Security built in coding and assembling
nSHIELD Environment security integration
Validate and Verify access to nSHIELD operatively
Implementation Process
Verification Process
Integration Process
Validation Process
Operation & maintenance
Operational readiness validation
Performance tests against security tests
Configuration management controls performance
SPD metrics based monitoring
Updates Upgrades management
Transition Process
Manteinance Process
Operation Process
Disposal
Develop disposal planning
Ensure information preservation
HW/SW disposal
Close systems
Disposal Process
Phase
Process
Activity

Figure 4-1: nSHIELD SPD Lifecycle support methodology based in NIST and SDLC of main ES manufacturers

The following subsection will define the plan for the methodology.

4.1 Initiation and training

4.1.1 Introduction

Initiation phase is about first, awareness and second, planning of the project so that first requirements can be gathered. Training in the field is a common activity in order to face the project with the correct instruments. Roles have to be determined and the nSHIELD system must be structured.

4.1.2 Activities

Activity #1.1 SPD Initial Planning

Description	Project engineers must be aware of the plan for accomplishing the successfully the entire engineering and operational task. The plan must reflect which stakeholders must be brought to the project in order to fulfil requirements in the correct way. Roles must be identified as well. This plan will depict the major milestones in the project
Expected Outputs	Plan report which will be dynamic Common understanding of security expectations
Synchronisation	Security concern issues must discussed in collaborative meetings between different stakeholders
Interdependencies	Each of the activities defined in this document should corroborate what the initial SPD plan specifies.
Implementer tips for its planning	
Security planning in the initiation phase should include preparations for the entire system life cycle, including the identification of key security milestones and deliverables, and tools and technologies. Many of the project initiation artefacts (meeting minutes, briefings, role identification) can be standardized and provided to developers for proper level-of-effort planning.	

Activity #1.2 Roles training

Description	This activity defines de roles that must be identified for training for completing the expected planning defined in previous activity for the success of the project.
Expected Outputs	Identification of Roles Assignment of trainings modules to roles
Synchronisation	Assignment between roles and training modules has to be balanced
Interdependencies	It is important to define clear dependencies between roles and assign correctly modules for each of the roles to be trained.
Implementer tips for its planning	
Key person selection is a very important milestone for the correct execution of the project. This activity is one of the most important ones for this reason. The assignment of the correct formation to different roles and the correct differentiation between modules is of paramount important.	

Activity #1.3 Categorise Information Systems

Description	This activity aims at inventorying the information system. Categorising information system is relatively difficult in security terms. The most important issues here is to define two types of taxonomies: 1- Taxonomy of types of information and information systems 2- Security Categories To categorise Information and information system types there are different standards and methodologies (i.e. RMM [09] from SEI). It consists in representing both information and systems information as actives of owned overall systems. There are several ways to categorise security. The conceptual way is to divide it in threat and vulnerabilities from one side and safeguard from another. In nSHIELD it is indispensable to categorise security vulnerabilities taking into account the layered architecture.
Expected Outputs	Taxonomy of information types and systems information types Security category for nSHIELD: Ontology derived from WP5
Synchronisation	There is a need of synchronisation with WP2 and WP5 (metrics and ontologies) that define somehow the scope of conceptual categorisations made in nSHIELD.
Interdependencies	Mapping between types of information and types of systems to security categories
Implementer tips for its planning	
nSHIELD provides mechanisms to categorise nSHIELD systems itself. Task 2.2. for defining metrics is an example and the work to develop Common Criteria ‘protection profile’, at least for the middleware and overlay will at least categorise different elements in both scopes identified during this activity.	

Activity #1.4 Assess business impact

Description	This is an activity that implements risks analysis and assess the impact of security threat and vulnerabilities in future system operation and business continuity. This activity provides the decisions instruments to those roles assigned to decide in the administration of systems (in this case nSHIELD system)
Expected Outputs	Contingency plan Business continuity plan
Synchronisation	Administration (operational) and engineering roles must be applied. This activity is synchronised with requirements identified by different uses cases in WP2.
Interdependencies	Information and systems categorisation must be applied
Implementer tips for its planning	
During roles assignment a person or group of people with business/operational oriented view should be targeted for this activity. This activity will focus on business continuity and convergence between security and business policies.	

Activity #1.5 Develop first nSHIELD requirements and metrics

Description	Specification of nSHIELD is the first attempt to describe the nSHIELD system formally. This activity which is divided in multiple sub-activities is one of the most important ones, due to the correctness of the specification is directly linked to the success of the result. nSHIELD will divide its specification in 3 areas: • SPD Requirements • SPD Metrics • SPD Architecture It is mandatory to use the key roles and persons to complete requirements from clients (final users) view (not only from technology point of view. For this nSHIELD will extract these requirements from main scenarios (4) described in WP7 and technology prototypes identified in technical WPs (3,4 and 5) The most important objective is to include Security, Privacy and Dependability built-in factors and attributes for the whole embedded systems lifecycle. This activity is, indeed, the starting point. The system requirements should reflect not only the core nSHIELD ones (SPD specifications), but also user specific requirements, such as business, operational, organizational, design, safety, validation and compliance standards.
Expected Outputs	nSHIELD specification based on real scenarios and divided in nSHIELD layers. Analysis of developed requirements will form consideration criteria during the acquisition process.
Synchronisation	There are many sub-activities that must be synchronised during this activity. Project leader has to deal with many factors in order to develop iterative but synchronised stages for this activity.
Interdependencies	This activity will be renewed during the whole project life time: it has interdependencies with all activities living during the design and operational phases of the security lifecycle. Indicatively, a strong interaction with the next phase exists (Development/Acquisition), since the criteria formed by the requirements analysis are used as input for several acquisition activities (development, purchase or enhancement of software).
Implementer tips for its planning	
Project leader must be a person with multiple views and flexible enough to manage the specification through all stakeholders as agile and successful as possible. Requirements should be use case or scenario based and led to uniformity and standardisation. In nSHIELD for example, common criteria could be used for profiling some software/hardware modules.	

4.2 Development/Acquisition

4.2.1 Introduction

Development is about implementing the engineering by your own and eventually obtaining a result which will be operational on some systems or sub-system.

Acquisition is the phase when you purchase a software/hardware module or a complete system, set it up and make it operational.

Both sub-phases are totally different from their origin point of view but have the same final goal: make the system be operational even if it is engineered in house or purchases/outsourced.

4.2.2 Activities

Activity #2.1 Define development process implementation

Description	A development process plan will be designed, according to the project's specificities and software life cycle model. During this process the methodology, tools, protocols, programming languages, hardware and software components will be defined.
Expected Outputs	Detailed plan for conducting the activities of the development process.
Synchronisation	This activity will be managed through the cooperation of nSHIELD WP2-WP7.
Interdependencies	The development of nSHIELD system incorporates the stages of work developed in setting the requirements and metrics, defining the architecture, and developing the different subsystems, as well as the interaction and feedback between these activities.
Implementer tips for its planning	
It is necessary that the operation and maintenance of products be dependent only on deliverable items, that is software and hardware that is adequately described during development process.	

Activity #2.2 Assess SPD risks to nSHIELD System

Description	This activity is about identifying and assessing security and dependability risks in nSHIELD Systems engineering. For doing that, both vulnerabilities and hazards will be identified and inventoried.
Expected Outputs	SPD risk analysis
Synchronisation	This activity will be managed through different risk analysis methods such as Magerit/ebios/ISAMM/ISF/ISO-IEC 13335/ISO-IEC17799/ISO-IEC 27001/Octave/Migra
Interdependencies	Risk analysis is dependent to security, privacy and dependability requirements gathering activity. It is also dependent with respect to the roles identification activity.
Implementer tips for its planning	
It is necessary to use tools for the risk assessment. The basic tool should be a checklist that ensures if all possible risks have been analysed and assessed against the system.	

Activity #2.3 Design SPD Reference Architecture

Description	For developing nSHIELD system, nSHIELD partners demand to construct a reference architecture which could be extrapolated to different domains and instantiated and linked to diverse legacy systems. This reference architecture is a continuous task that must be improved together with the requirements gathering stage. Definition and specification of reference architecture is and will be the main outcome of nSHIELD. WP2 defines a whole task (with two deliverables) for this activity.
Expected Outputs	Reference architecture (2 iterations)
Synchronisation	There must be a strong link between the reference architecture and work done in technical WPs such as Wp3, 4 and 5. In these WPs, Layers of nSHIELD system must specify what is identified and structured in the reference architecture.
Interdependencies	It has strong dependencies with requirements and modular design activities: components design.
Implementer tips for its planning	
An architecture designer should be present at this stage in order to measure the correct detail and granularity of the nSHIELD architecture. Project leader should delegate his/her authority to architecture designer and let him/her coordinate this activity with the rest of the agents involved in the project.	

Activity #2.4 Develop SPD Controls and Metrics

Description	Both architecture and requirements need to be controlled and measured somehow in order to explicitly define its boundaries. This activity is about including these controls within the engineering process so that security, privacy and dependability can be measured quantitatively.
Expected Outputs	Controls and metrics and their links to be enforced during operational phase
Synchronisation	This activity is implemented during task 2.2 in nSHIELD and will be enforced in WP6.
Interdependencies	Strong relationship with formal and scenario based requirements which are extracted from task 2.1 (requirements) and from WP7 – Scenarios.
Implementer tips for its planning	
nSHIELD, as an innovative project, is not only developing metrics and controls for each of SPD attributes for each layer but also the overall view of SPD. For doing that, a composition of metrics and controls will be developed. It is important to develop a practical compositional approach of metrics (not too academic) so we can deploy this solution in a real or semi real environment.	

Activity #2.5 Implement documentation and testing plans

Description	While the most prominent document is the System Security Plan, nSHIELD documentation supporting it may include: • Configuration management plan • Contingency plan (including a Business Impact Assessment) • Continuous monitoring plan • Security awareness, training and education • Privacy impact assessment (PIA) For evaluating and testing systems software and hardware a plan must be developed defining the type of test, stakeholders involved and modules resulted from these tests. These tests must be documented as well.
Expected Outputs	Support documentation and test plan
Synchronisation	This documentation should be developed during WP2, WP3, WP4 and WP5 and implemented in WP6 and WP7.
Interdependencies	This documentation has a strong link with previous work done in activities such as requirements definition, metrics and control identification and reference architecture structure implementation.
Implementer tips for its planning	
Documentation, although heavy, should be a formal task within the engineering process. It is important to implement it correctly because although the role always remains, people usually leave from the project and other person shall incorporate. For a rapid inclusion it is important a standard approach for the documentation so that new person approaching the project can easily understand what is being done. Preliminary testing may be done at component or security zone level to ensure that each component or security zone is secure as an entity. Source code should be periodically reviewed using automated tools or manual spot check for common programming errors that have a detrimental impact on system security	

Activity #2.6 Prepare an acquisition plan

Description	The acquirer should prepare, document and execute an acquisition plan. The plan should include the following: • System requirements • Planned employment of the system • Type of contract to be employed • Responsibilities of the organizations involved • Support concept to be used • Risks considered as well as methods to manage the risks
Expected Outputs	Definition of product selection, acceptance and acquisition strategy.

Synchronisation	The activity concerns the external user, but may be handled over to an agent representing him/her.
Interdependencies	From nSHIELD documentation, the activity has a strong link with the Requirements definition.
Implementer tips for its planning	
It is important that the acquisition methodology is systematized. This is the way to ensure that the acquired product will fulfil client needs and simplify lifecycle support.	

Activity #2.7 Control acquisition contracts

Description	Acquisition is a large activity which could be divided into different sub-activities. Acquisition comprises the adoption of external software and hardware (even a complete system) which is intended to perform the same operational functionalities (as if it were designed by your own). Acquisition includes as well a contract between provider and client that must be managed. This contract will be followed up by the client requiring all stipulated requirements for the correct result of the deployed modules. A series of actions outline the procedure of contract preparation. Firstly, the client establishes a procedure for supplier selection, a point where nSHIELD platform should present a system proposal that will succeed the evaluation criteria. After the selection of nSHIELD product, the contract's negotiation phase begins, where important parameters should be included, addressing requirements, services and costs. Finally, the procedure for amendments should be defined, covering the need for possible changes along with the impact that these changes would have on the project's scheduling and lifecycle.
Expected Outputs	Contract and its management plan. The contract should address proprietary, usage, ownership, warranty and licensing rights associated with the reusable off-the-shelf software products.
Synchronisation	This activity is quite out of the scope of nSHIELD. nSHIELD defines the architecture to complement new nSHIELD bases nSHIELD systems with legacy systems but does not explicitly express required requirements with those new purchased components and sub-systems.
Interdependencies	It shall require dependencies with systems requirements and business impact activities.
Implementer tips for its planning	
The role of business impact measurer shall be present at this stage to measure the impact of the entrance of new software/hardware into nSHIELD systems. It is important to measure security interoperability in here and therefore nSHIELD SPD Gateway should be developed taking into account this activity.	

4.3 Implementation & assessment

4.3.1 Introduction

Implementation and assessment is the tangible result of the design: what is designed needs to be implemented for make real its operational view. All software and hardware modules will be validated via tests and validations and verification activities.

4.3.2 Activities

Activity #3.1 SPD Implementation planning

Description	What is designed during the previous phases has to be implemented. This activity is about planning the implementation phase. Planning should encompass persons in charge to code the source code and develop hardware prototypes for after that develop test cases for validating them and eventually develop an industrialisation plan.
Expected Outputs	nSHIELD SDP implementation plan
Synchronisation	It is synchronised with each of the implementation plans of technical WPs (3, 4 and 5) and the integration plan of WP6. It shall be linked also with the metrics algorithms develop between WP2 and 6.
Interdependencies	This activity has strong link with designed components in previous activities. Indeed, it is an iterative and improvement process that goes from implementation to design phases.
Implementer tips for its planning	
Below the project leader, there should be a person in charge to coordinate the implementation phase that is different from the architecture designer but still has to be coordinated with him. This person will assume the implementation task and has to be present for developing this plan together with the project leader.	

Activity #3.2 Security, privacy and dependability built in coding and assembling

Description	The basis for developing security-by-default and by-design factors is to incorporate security and dependability factors to the coding stage. This activity is to prepare and perform security patterns and to intrinsically comprise them at coding/implementation stage.
Expected Outputs	Verifiable security, privacy and dependability coding and implementation patterns that could be tested in later activities
Synchronisation	This activity shall be linked to WP3, WP34 and WP5 implementation prototypes and the integration stage in WP6.
Interdependencies	It has an iterative link towards the implementation plan and will recursively interact with design phase activities
Implementer tips for its planning	
A good software/hardware analyst/programmer is key to the correctness and success of the engineering	

process. There are many tips to take into account for a secure and dependable coding of software and robust implementation of hardware. It is usually been performed under the experience of the engineer. However, in nSHIELD, there will be a uniformity undergoing task that will look into different secure coding methodologies (SEI) and standards (ISO27002). Furthermore, nSHIELD SPD coding process will bring a formal approach.

Activity #3.3 nSHIELD Environment Security Integration

Description	nSHIELD objective is to develop a reference architecture that will make security, privacy and dependability interoperable and embeddable in different and heterogeneous domains and with other legacy systems. nSHIELD presents for that a whole WP6 (Integration). This activity will also encompass validation process for each of the integrated modules or sub-systems and the systems as a whole.
Expected Outputs	nSHIELD system
Synchronisation	As stated, a whole WP (6) is defined for this task in nSHIELD project.
Interdependencies	This activity depends on WP2 requirements, design phases like reference architecture and the implementation process.
Implementer tips for its planning	
Integration usually is the bottleneck of big project such as nSHIELD. Different modules corresponding to different components in the reference architecture should be exposed and integrated. Not only these components should be integrated but also the integration with existing systems of systems should be demonstrated. nSHIELD gateways should be validated at this stage and all functionalities (integration from the atomic to the most complex components and functionalities) should be verified at this stage.	

Activity #3.4 Validate and verify access to nSHIELD operatively

Description	Validation and verification process is a formal activity that provides indeed formality and nearby certification approach to the solution outcome from the implementation phase. The first approach is to validate that all functional and SPD tests have been correctly implemented and eventually verify that they are correctly being proved in the real scenarios. nSHIELD should encompass both validation and verification sub-phases.
Expected Outputs	Validation report and verification means and proofs.
Synchronisation	This activity is directly linked to WP6 (after integrations) and WP7 about verifying nSHIELD outcomes in real scenarios (4 scenarios identified in nSHIELD:
Interdependencies	It has a strong dependence with testing plan developed in previous stages.
Implementer tips for its planning	
There should be a quality tester and moreover in nSHIELD project shall be a SPD tester for this activity. SPD tester should be in touch with SPD scenarios owners in order to prove all potential SPD functionalities of nSHIELD Systems.	

4.4 Operation and maintenance

4.4.1 Introduction

Once the engineering phase has been addressed, the operation phase comes. In the operational approach, systems monitoring and measuring is mandatory in order to know the current status of the system. An additional difficulty is the updates/upgrades managements which usually is a weakness and makes systems vulnerable. During the functional cycle of the product the supplier provides operational support to the user. All these and more issues must be addressed during this phase which is that phase for monetizing the system.

4.4.2 Activities

Activity #4.1 Operational readiness validation

Description	Many times when a system transitions to a production environment, unplanned modifications to the system occur. If changes are significant, a modified test of security controls, such as configurations, may be needed to ensure the integrity of the security controls. This step is not always needed; however, it should be considered to help mitigate risk and efficiently address last-minute surprises.
Expected Outputs	Evaluation and assessment with respect unexpected changes
Synchronisation	WP7 should resolve unexpected changes.
Interdependencies	Linked to tests developed in validation and verification atomic activities.
Implementer tips for its planning	
System administrator should elaborate a historic of different malfunctions derived from changes. This should be sent to system developers in order to develop new configurations for making this unexpected happenings more embeddable to the overall system performance	

Activity #4.2 Performance tests against security tests

Description	This activity measures the SPD impact on the overall system. There always has been a trade-off between security and performance: while inserting more security into the systems less performance and vice versa. This activity will measure the balance between them.
Expected Outputs	Balancing test
Synchronisation	Balancing test shall be developed in WP6 and WP7 with the support of metrics module.
Interdependencies	It has a strong dependence with business impact activity. It validates the plan developed in that stage.
Implementer tips for its planning	
Both roles technical and business should be incorporated to this activity in order to fix a balancing threshold for nSHIELD operational system.	

Activity #4.3 Configuration management controls performance

Description	An effective agency configuration management and control policy and associated procedures are essential to ensure adequate consideration of the potential security impacts due to specific changes to an information system or its surrounding environment. Configuration management and control procedures are critical to establishing an initial baseline of hardware, software, and firmware components for the information system and subsequently for controlling and maintaining an accurate inventory of any changes to the system. Changes to the hardware, software, or firmware of a system can have a significant security impact.
Expected Outputs	Change control decision and updated configuration document
Synchronisation	Linked to WP1 (decisions documents) and WP7
Interdependencies	Security architecture should provide key details on component-level security service, which in turn provides a benchmark to evaluate the impact of the planned change.
Implementer tips for its planning	
In the configuration management many person could be involved (normally final users using the systems.) However, it will turn that the systems administrator will be the person in charge to manage controls of configuration management.	

Activity #4.4 SPD metrics based monitoring

Description	Monitoring is a useful mechanism in order to know the status of the system. nSHIELD will provide a SPD metrics based monitoring tool. This monitoring tool will detail each of the metrics defining the boundaries of the systems and the aggregation of them.
Expected Outputs	Monitoring tool/method
Synchronisation	This activity is synchronised with WP5 (semantic layer) and WP2 metrics definition.
Interdependencies	Clear dependencies with how the monitoring module in the architecture has been defined for operational view
Implementer tips for its planning	
System administrator view has to be taken into account for the definition of monitoring method.	

Activity #4.5 Updates and upgrades monitoring

Description	Updates and upgrades often demand a great effort to manage them. There several weaknesses in the process itself and risk when upgrading must be considered. System administrator usually performs lots of tests for validating massive updated and upgrades. It comes critical when upgrading firmware from embedded systems that are critical systems.
Expected Outputs	Updates/upgrades method
Synchronisation	It shall be linked to WP7 scenario based SPD upgrades
Interdependencies	It has strong dependency with configuration management, requirements change when upgrading and updating
Implementer tips for its planning	
All updates and upgrades should be validated and verified before moving them to production phase.	

Activity #4.6 User support

Description	The supplier shall establish a procedure for SPD product operational support. A plan to handle problems will be set, in order to receive, record and resolve issues. The provider will also accept client requests and take corresponding actions or provide assistance and consultation.
Expected Outputs	Reported problems will be handled. When feasible, the supplier waiting to reach a viable permanent solution shall provide a work-around method to temporarily serve client needs.
Synchronisation	User support is linked to WP7 applications and to the technical development of nSHIELD system.
Interdependencies	The activity has strong interdependencies with the application and market perspectives of nSHIELD.
Implementer tips for its planning	
Support plan shall be documented and executed.	

Activity #4.7 Maintenance

Description	The objectives of the maintenance process during lifecycle support include: • Smooth transition after modifications • Preservation of system integrity • Migration to a new operational environment • Retirement of active support The product shall undergo necessary update and improvement procedures without endangering its integrity. The migration plans shall include development

	of tools, conversion of software and migration verification (after its execution). The process of partial or full support retirement shall be gradual, organized and smooth, accompanied by activities that ensure viability, such as possible transition to the new software product, responsibility for any future residual support issues and accessibility of archive copies of data.
Expected Outputs	Product maintenance is systematized.
Synchronisation	Maintenance is linked with a commercial product development and thus indirectly linked to technical development of WP3-WP7.
Interdependencies	The activity has strong interdependencies with the application and market perspectives of nSHIELD.
Implementer tips for its planning	
Maintenance plan shall be documented and executed.	

4.5 Disposal

4.5.1 Introduction

Disposal, the final phase in the SDLC, provides for disposal of a system and closeout of any contracts in place. Information security issues associated with information and system disposal should be addressed explicitly. When information systems are transferred, become obsolete, or are no longer usable, it is important to ensure that government resources and assets are protected.

4.5.2 Activities

Activity #5.1 Develop disposal planning

Description	Disposal plan for the correct close out of nSHIELD system
Expected Outputs	Disposal plan
Synchronisation	WP7 should define in the scenario
Interdependencies	Last checkout with last planned requirements.
Implementer tips for its planning	
Disposal should be developed by system administrator and new leader of next project/system developer.	

Activity #5.2 Ensure information preservation

Description	This activity deal with the correct transfer of valid information to new systems.
Expected Outputs	Secured data flow
Synchronisation	WP7 should define in the scenario
Interdependencies	It is defined by Business impact document

Implementer tips for its planning
Information preservation should be developed by system administrator and new leader of next project/system developer.

Activity #5.3 HW/SW disposal

Description	Hardware and software can be sold, given away, or discarded as provided by applicable law or regulation. The disposal of software should comply with license or other agreements with the developer and with government regulations.
Expected Outputs	Disposition records for hardware and software.
Synchronisation	Updating of system and component inventories in WP7
Interdependencies	It is defined by Business impact document
Implementer tips for its planning	
Hardware/software disposal should be developed by system administrator and new leader of next project/system developer.	

Activity #5.5 Closing systems

Description	The information system is formally shut down and disassembled at this point.
Expected Outputs	Documentation verifying system closure
Synchronisation	-
Interdependencies	Final review by Business impact document
Implementer tips for its planning	
A memorandum articulating formal system closure and proper action taken that includes in the distribution all key stakeholders provides the simplest approach to formal closure.	

5 Plan execution approach

There is one approach that explicitly did not be mentioned within this document. It is the industrialisation process which encompasses the phases between implementation and operation. The plan execution approach for developing the report will start at this stage: when the design and implementation parts of nSHIELD have concluded (during the nSHIELD project) and the operation (also industrialisation) phase starts so that we can define a whole SPD lifecycle support for this.

Therefore, the plan execution will start for operation phase guaranteeing that the proposed architecture to be future proof, to support the installation, download and upgrade cycle of nSHIELD representing SoSs. Furthermore, this plan aims at addressing security and integrity without comprising performance and cost when it is deployed.

Validation and verification processes should be extended to SoS operational lifecycle support and thus, different tests will be planned in order to view the current status of the system.

So that, what we are planning is how to proceed developing a methodology for the operational view of nSHIELD System. The plan will encompass the activities within phases described in previous section and will describe how nSHIELD will internalise these activities.

For each of the selected activities the following elements will be described and instantiated for nSHIELD system:

- Introduction: defines a brief overview of the activity selected. It defines the objectives and the scope.
- Inputs and outputs of the activity: artefact needed for processing inputs for generating outputs. Identification of these artefacts.
- SPD by X (default, design) process implementation tips for this activity
- Synchronisation towards nSHIELD system. How nSHIELD system analyses this activity
- Relations towards diverse standards
- Measurements: from SPD goals to SPD evidences
- Guide for applicants

These elements will be a chapter or section of nSHIELD SPD Lifecycle Methodology.

In parallel and optionally a process defining multiple activities could be described. This process will be similar to the structure of processes that ISOs define. The process could encompass the following elements:

- Terms and definitions
- Application of this process
- Relationship between nSHIELD systems and HW/SW elements
- Organisations and parties
- Organisation level and project level adoption
- Description of process
- Decomposition of the process
- SPD life-cycle model and stages

- Technical definition
- Outcomes

Both activities and processes compose a phase. The overall objective is to describe these elements as deep as possible in order to complete a phase based methodology for defining nSHIELD SPD based System.

The activities that are planned to be described more detailed. Industrialization phase is one of the phases that will be argued by partners for deliverable 6.4. It is an important phase that has to be supported by an exploitation plan. Therefore it is planed that during the execution of this plan that all partners argue about how to incorporate **industrialization** phase (activities and processes) to this plan.

6 Conclusion

This document defines a plan for decomposing SPD activities during design and operation phases for nSHIELD.

The plan does not define a timeline because it is a plan for defining a methodology: the operational method for using nSHIELD System. It should be delivered as a guideline for administrators and other different roles across an industrial organisation.

The execution of the plan will be delivered in D6.4 Lifecycle and SPD support report. This deliverable will be defined by the different planned activities identified in this document. However, these activities might vary and we could focus on some of them and not in all of them due to the importance of systems security, privacy and dependability engineering in nSHIELD.

7 References

[01] ISO12207-2008: http://www.iso.org/iso/catalogue_detail?csnumber=43447

[02] ISO 15288: 2008: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43564

[03] Microsoft Security Lifecycle : http://www.microsoft.com/security/sdl/default.aspx

[04] http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP800-64-Revision2.pdf

[05] NIST Security Lifecycle: http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP800-64-Revision2.pdf

[06] Cisco CSDL: http://www.cisco.com/web/about/security/cspo/csdl/index.html

[07] Security Engineering, A guide to building dependable distributed systems, Ross Anderson. Wiley. 2001

[08] http://www.microsoft.com/en-us/download/confirmation.aspx?id=12285

[09] www.cert.org/archive/pdf/10tr012.pdf