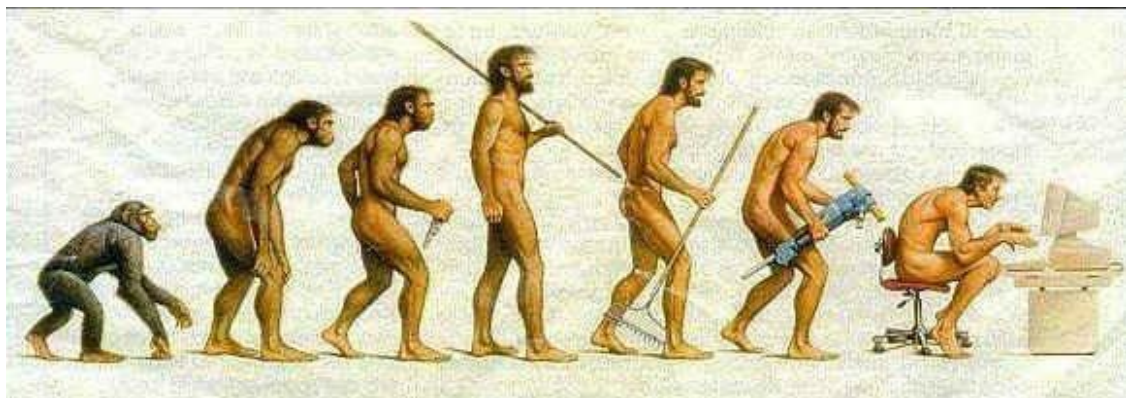




# SERIOUS

## DELIVERABLE

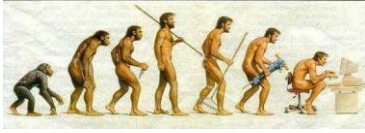
### D3.5 – User needs for security case study



Project number: ITEA 04032  
Document version no.: WP3 Deliverable 3.5 Final version  
Edited by: Softeam, February 19<sup>th</sup>, 2007

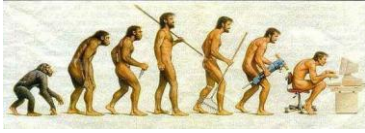
**ITEA Roadmap domains:**  
Major: Services & software creation

**ITEA Roadmap categories:**  
Major: Software engineering  
Minor: System engineering



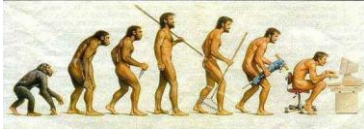
## HISTORY

Document version #	Date	Remarks
V0.1		Starting version, template
V0.2	21/09/2006	Definition of ToC
V0.3	13/12/2006	Draft version, contributions by partners
V0.4	19/02/2007	Remarks from Surlog
V1.0	26/02/2007	Final Version (Approved by PCC)



## TABLE OF CONTENTS

<b>1. GOAL OF THIS DOCUMENT .....</b>	<b>4</b>
<b>2. CONTEXT .....</b>	<b>5</b>
2.1. Expression of security needs .....	5
2.2. Study of threats .....	6
2.3. Expression of security objectives .....	7
<b>3. SECURITY NOTIONS .....</b>	<b>8</b>
3.1. Availability .....	8
3.2. Access control.....	8
3.3. Data integrity.....	8
3.4. Data confidentiality.....	9
<b>4. USE CASES.....</b>	<b>10</b>
4.1. Introduction .....	10
4.2. Actors.....	10
4.3. The “Audit” use case .....	10
4.4. The “Generate a report” use case .....	11
4.5. The “Annotate the UML model” use case.....	11
4.6. The “Improve security” use case .....	12



## 1. Goal of this document

SERIOUS aims at improving techniques to smoothly integrate software analysis and software refactoring techniques included in the day-to-day software process, leading to a truly evolutionary software-engineering model. Possible directions for solutions are service oriented architectures (SOA's) and trend analysis of the volatility of software. The former means the software infrastructure has been designed taking evolution into account. The latter means the direction of software growth is made clear, during all phases of the development process, enabling possibilities to clearly manage it.

SERIOUS aims to define various quality aspects in such detail that they can be applied and used in the early phases of software development and in the evolution phase.

The objective of work-package 3 is to provide methods, tools, models and platforms for quality control in software evolution.

The aim of 3.4 task of the third work package is to be able to synthesise a security model from existing source code, cast in UML syntax. Moreover, this model has to be extended with security properties and/or threats inferred from this source code. It is important to determine which kind of properties (metrics) are relevant in the context of security and can be statically extracted from a source code. Based on this set of properties, an analyser will be developed to automate the detection of possible threats in the software source.

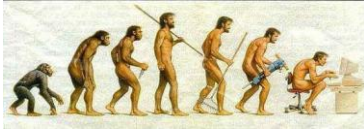
Security is a key quality attribute of software and services. In many cases it is one of the main forces for evolution (in order to face new threats). Security reference models, reference architecture, security patterns will be defined and specific architecture refactoring activities will be defined for it.

An UML modelling tool, able to support security meta-model and patterns will be developed in this task and applied in the aforementioned case studies. Task 3.4 will elaborate a framework for security design and development from existing software through:

- Definition of security metrics
- Definition of patterns (security policies using UML model)
- Enhancement of UML model with security analysis

The aim of the WP 3.4 task is to elaborate the framework basis to later (WP3.5) synthesize an automated tool extracting a security model from existing source code. This model, expressed in UML syntax, will represent the source code architecture. Moreover, it will be extended with safety/security properties and/or threats inferred from this source code.

The purpose of the document is to list the user needs related to the security of an application.



## 2. Context

An information system is based on the essential elements, functionalities and information that make up the added value of the organisation's information system.

Software applications play a strategic role in all management operations at the financial and economic levels, as well as in systems where human life is at stake (transport, arms system, and so on). Their omnipresence, together with the importance of the services they provide, makes their security a critical issue for end-users and for almost all organisations. These factors combine, with the ever-increasing complexity of information systems to further increase, the critical nature of software security. Security issues concern practically all organisations, with financial systems and systems where human lives are at stake being particularly conscious of their importance.

In the security domain, different types of commercial party play a role:

- Antivirus editors
- Firewall editors
- Web application firewall editors
- Strong authentication solution editors
- Anti-spam solution editors
- Encryption solution editors
- Security Appliance Constructors for detecting and preventing intrusion
- Editors of operating systems based on security: OpenBSD, SELinux
- IT engineering and service provider companies, specialised in security issues
- MSSPs: Managed Security Services Providers

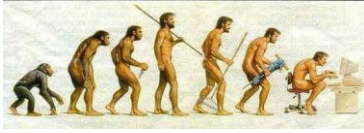
*For example, a rocket launch trajectory control system relies on a wide range of information, such as calculation parameters and results, as well as the different functionalities that allow these calculations to be made.*

Essential elements are linked to a set of entities of various different types: materials, software, networks, organisations, personnel and sites.

*For example, a parameter used in a rocket launch trajectory calculation is linked to control computers, processing software, operators, the rocket itself, domain regulations, and so on.*

### 2.1. Expression of security needs

**Information system security** refers to the complete set of essential technical, organisational, legal and human means in place to conserve, re-establish and guarantee the security of the information and the information system.



Each essential element has a need for security in order for the business to function correctly.

This need for security is expressed using different security criteria, such as availability, integrity and confidentiality. If a security need is not met, the organisation can be affected in several different ways: financial loss, decreased efficiency in the running of the business, negative impact on brand image, danger to personnel, pollution, and so on.

*Once again, let's consider the example of a parameter used in a rocket launch trajectory calculation. This information should have a strong need for availability and integrity, in order to avoid any potential danger to the safety of personnel.*

## **2.2. Study of threats**

Furthermore, every organisation is exposed to various threatening elements, in terms of its natural environment, its culture, its image, its domain, and so on.

A threatening element is characterised by its type (natural, human or environmental) and its cause (accidental or deliberate).

A threatening element can use different methods of attack, which must be identified.

A method of attack is characterised by the security criteria (availability, integrity, confidentiality...) it can affect and by the threatening elements likely to use it.

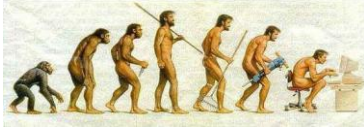
*Let's continue with our example. For a rocket launch organisation, a large number of methods of attack and threatening elements must be taken into account:*

- *physical accidents (fires)*
- *natural events (seismic phenomena)*
- *loss of essential services (loss of electricity supply)*
- *compromised information (software trapping)*
- *technical failure (material malfunctioning)*
- *physical attacks (sabotage)*
- *errors (interpretation errors)...*

Every entity has specific weaknesses that could be exploited by threatening elements according to each method of attack.

*Thus, several vulnerabilities linked with the rocket launch organisation can be identified:*

- *the possibility of the existence of hidden functionalities introduced during the design or development phases (software)*
- *the use of non-evaluated materials (materials)*
- *the possibility of creating or modifying system commands (networks)*



- *the possibility of acting on system resource software via the network (networks)*
- *the possibility of entering the site via indirect access points (premises)*
- *the non-respect of orders on the part of certain operators (personnel)*
- *the absence of security measures during the design, installation and utilisation phases (organisation)...*

### **2.3. Expression of security objectives**

Security objectives are the expression of the intent to counter identified risks and/or to satisfy organisational security policies. An objective can concern the target system, its development environment or its operational environment. These objectives can then be further defined as security functionalities that can be implemented on the information system.

All that remains then is to determine how essential elements can be affected by threatening elements and by their methods of attack: this is the concept of risk.

Risk represents a possible source of damage. It is the fact that a threatening element can affect essential elements by exploiting the vulnerability of the entities on which they rely using a particular method of attack.

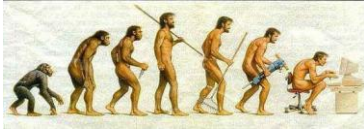
In our example, a risk consists of the compromise of sensitive information through software trapping, due to the possibility of creating or modifying system commands linked to the network, which could have an impact on personnel security and brand image.

Security objectives mainly consist in covering the vulnerability of the entities that make up the set of retained risks.

There is no point in protecting what is not in danger. Furthermore, the greater the potential for attack, the higher the level of security objectives.

Thus, these objectives constitute a perfectly adapted specification document.

*One of the security objectives for the rocket launch organisation is to secure the creation and modification of system commands linked to the network.*



## 3. Security notions

### 3.1. Availability

Information and services are accessible in a reliable way when they are needed. The information transmitted on the Internet must be transmitted by means of secure channels and reliable storage equipment, which are operational when they are needed.

IT equipment must be protected not only against physical damage, but also against power cuts, system failures and overloads.

Data saves, monitoring and the use of antivirus software and adequate IT resources are all security measures that ensure the availability of data and services.

Availability is strengthened by material and software architectures providing the following characteristics:

- Clustering
- Data availability
- Local material redundancy
- Load balancing
- Multi-site redundancy

### 3.2. Access control

Access control is a security service that aims to check the legitimacy of a subject's access (user or software, for example) to an object (file, printer, ...) in the aim of carrying out a given operation (reading, writing, ...).

Access control allows the limitation of the actions or operations that a user or an IT system can realise.

Therefore, it restricts what the user can do directly, but also the programmes run upon his request, in order that only authorised entities be able to access information system resources.

The aim is to avoid the appearance of a security problem.

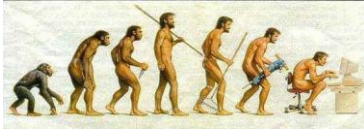
Authentication also allows user identity to be checked.

### 3.3. Data integrity

Data integrity is a set of measures aimed at avoiding the intentional or accidental loss or alteration of data, and also a group of techniques and means used to check and prove that data has not been altered.

For example, comparing an original and a copy allows the non-alteration of data to be proven.

Possible dangers are as follows:



- Loss of data through the destruction or alteration of the material (voluntary or involuntary destruction, damage caused by natural elements, ...)
- Destruction or falsification following incorrect processing (involuntary destruction by the user, material malfunctioning, software errors, ...)
- Impossibility of accessing data due to an access mechanism problem (loss of access software, fault or non-availability of material, personnel error, ...)

Possible security measures are as follows:

- Security copies (different material, decentralisation, operation log, ...)
- Physical protection (against water, fire, magnetism)
- User training
- Documentation on the material and software
- Limitation of access to the systems and to copies

Validity checks contribute to ensuring the semantic consistency of the database. They must prevent the introduction of non-compliant data into the database (negative age, 13<sup>th</sup> month of the year, and so on). These checks are also used to verify that the database has not been subjected to this type of alteration.

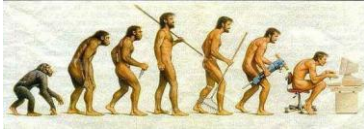
### **3.4. Data confidentiality**

Data confidentiality is a concept used to ensure that information can only be read by authorised people.

Encryption is one of the elements in the message coding and decoding process that ensures data confidentiality. The exact name of this process is "*cryptology*", which consists in transforming data using mathematical formulae to protect it.

Cryptology is first and foremost a software solution that should not require any major investment in terms of hardware. It is an essential tool to ensure data confidentiality, since only authorised people can view the data in question. It also guarantees data integrity, since it protects data against unauthorised modifications and alterations.

The cryptographic methods used are known for their robustness, which makes it easy to control the risk of any unauthorised decoding.



## 4. Use cases

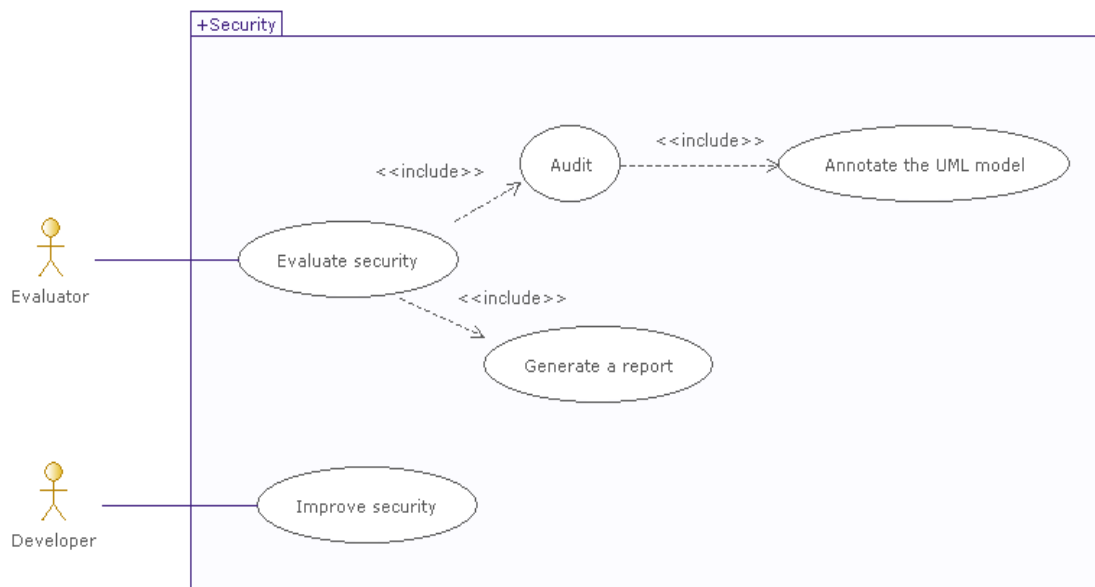
### 4.1. Introduction

In this chapter, we are going to concentrate on the "software" part of security.

The aim of the "software" part is to provide services that will allow the security of an application's code to be analysed, in order to subsequently improve it.

We are going to use a model-based approach. This will allow security to be better mastered by studying its characteristics and designing them with a higher level of abstraction than the one provided by the code. The UML model has been chosen as the basic modelling language due to its widespread recognition throughout the software industry.

### 4.2. Actors



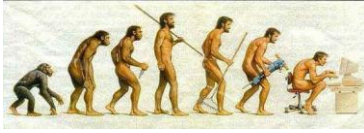
The involved actors are:

- Developer: person who develops the application
- Evaluator: person or system that analyses the application, in order to define its level of security

### 4.3. The “Audit” use case

The “Evaluator” triggers the audit of the UML model to define the application's security level.

This operation can be triggered "at design time" to define an application with a high level of security, or on an existing application that has already been



coded, in order to audit its security level and potentially reinforce it. This operation can be run on the UML reverse engineering of applicative code, or on the application model, where this exists.

The audit can consist of various static analyses of the code, in order to detect possible security risks.

The audit can aggregate and summarise different security criteria using UML structuring mechanisms, notably packages. Model elements can be sorted and presented according to their risk level, by type of security criteria. Typically, the UML units on which security criteria are evaluated are classes, packages and operations.

The aim of the audit is to identify the UML units for which the threat is significant, as well as those that present a particular risk.

The UML model can be architecturally designed and structured in order to distinguish the "external" parts, whose characteristics are accessible to an external participant, from the "internal" parts, which can only be accessed by the application itself. Layered architecture and careful management of encapsulation promote more efficient security management.

Access control management must be identified through dedicated UML packages, separate from the applicative part. Similarly, data protection through encryption must be based on identified services. Data received from or sent to the outside must be encrypted using these services, where appropriate.

#### **4.4. The "Generate a report" use case**

The "Evaluator" triggers the generation of a report on the level of security.

The report includes sections automatically inferred from annotations made to the model, as well as remarks written by the Evaluator.

This report is intended for the Evaluator himself, who will use it to ensure that his work is complete, as well as for those responsible for the application, who then have a security audit of their system.

#### **4.5. The "Annotate the UML model" use case**

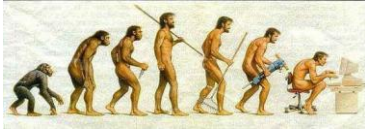
The "Evaluator" annotates the UML model using extensions defined in a UML profile, in order to make security notions appear in the UML model.

The aim of these annotations is to use traceability between the different levels of abstraction and to allow more straightforward security certification.

Thus, security requirement traceability will be realised from the needs expression document right through to the code.

It is also possible to describe the potential attacks by human actors or by other systems, as well as the reactions of the system to these threats.

In some cases, these annotations will allow specific code to be automatically generated, for example to check that the system is secure, to provide security services or to make components more secure, thereby facilitating the work of the "Developer".



#### **4.6. The “Improve security” use case**

The “Developer” improves the security of the software based on the report provided by the Audit.

This improvement can require:

- Modification of the UML model for security integration and reliability
- Implementation of data access checks
- Encryption of data to ensure confidentiality
- ...

The architecture of the software often needs revision, in order to separate visible parts from internal parts, improve encapsulation and introduce encryption and data access control services in the appropriate places.