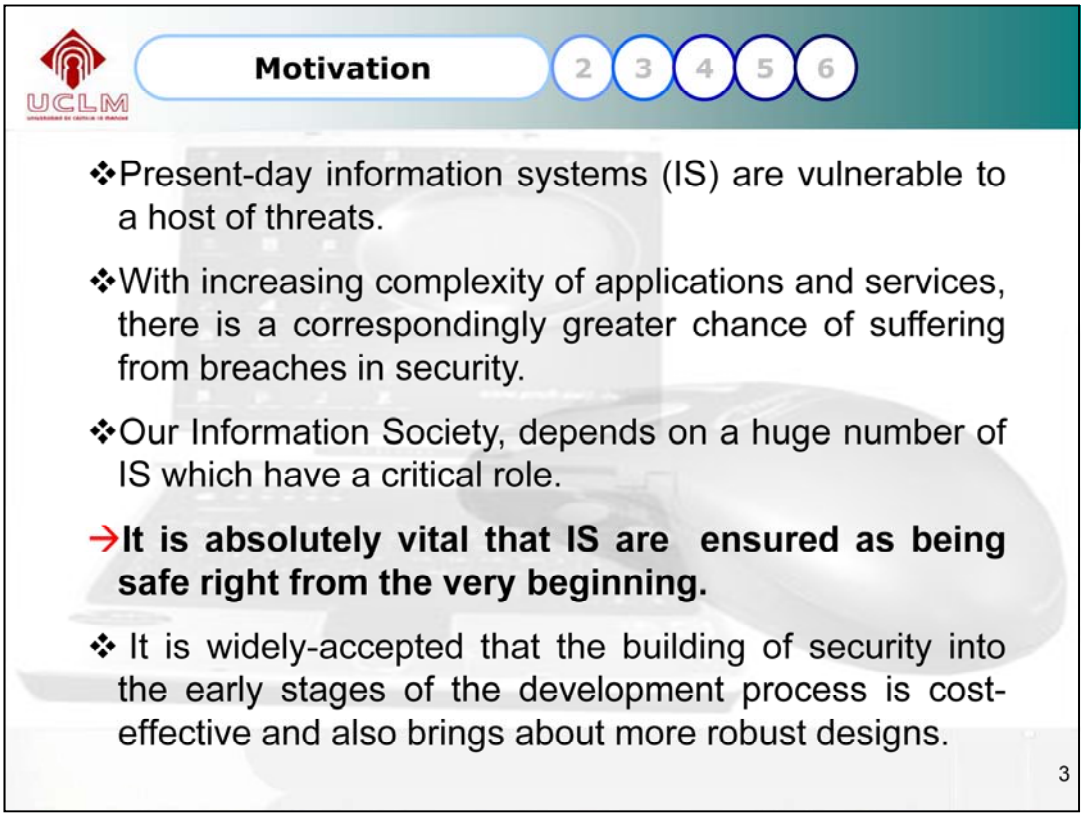
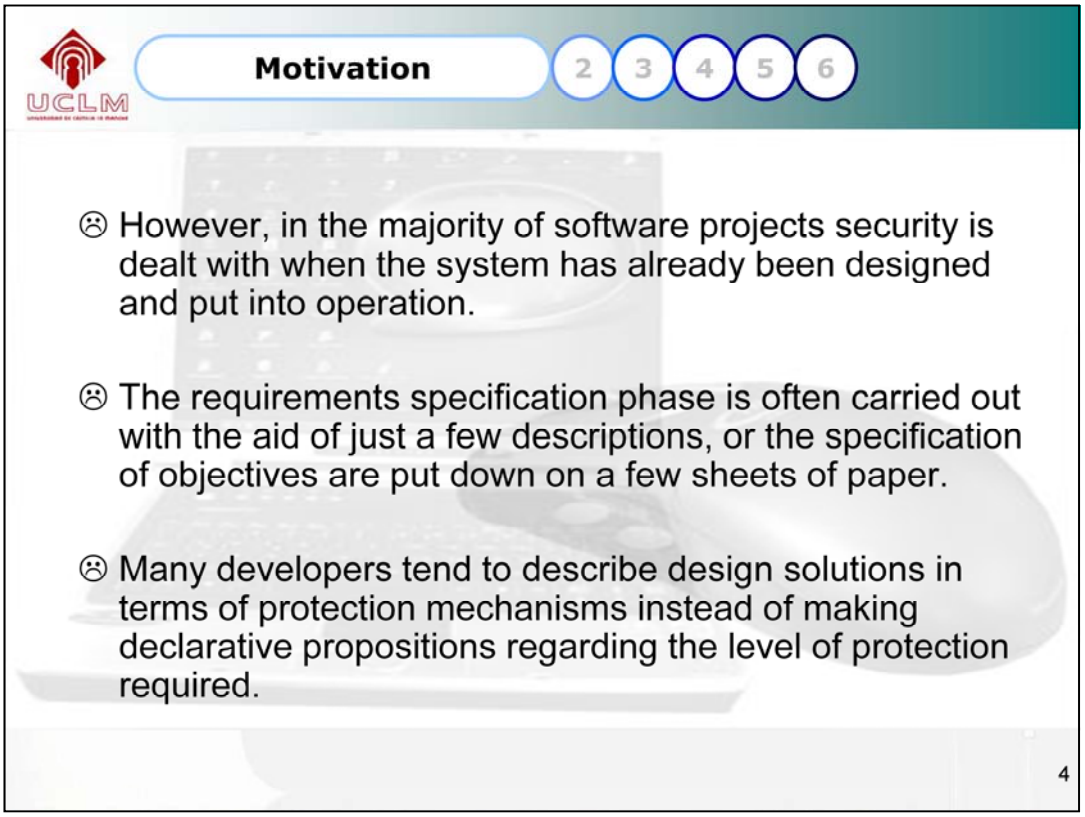

15 May, Trento (Italy)
© GSyA Research Group
SREP
Security Requirements Engineering Process
Speaker:
David Garcia Rosado
GSyA Research Group, University of Castilla La-Mancha
(Spain)
Authors:
Dr. Daniel Mellado Fernández
GSyA Research Group, University of Castilla La-Mancha
(Spain)
Dr. Eduardo Fernández-Medina
GSyA Research Group, University of Castilla La-Mancha
(Spain)
and
Dr. Mario Piattini
Alarcos Research Group, University of Castilla La-Mancha
(Spain)
© Images are property of Gerencia de Informática de la Seguridad Social and Instituto Nacional de la Seguridad Social
UCLM

CONTENTS



•Present-day information systems are vulnerable to a host of threats. What is more, with increasing complexity of applications and services, there is a correspondingly greater chance of suffering from breaches in security. In our contemporary Information Society, depending as it does on a huge number of software systems which have a critical role, → it is absolutely vital that IS are ensured as being safe right from the very beginning.



•The biggest problem, however, is that in the majority of software projects security is dealt with when the system has already been designed and put into operation, that is, the security requirements are undervalued. Added to this, the actual security requirements themselves are often not well understood. This being so, even when there is an attempt to define security requirements, many developers tend to describe design solutions in terms of protection mechanisms, instead of making declarative propositions regarding the level of protection required.

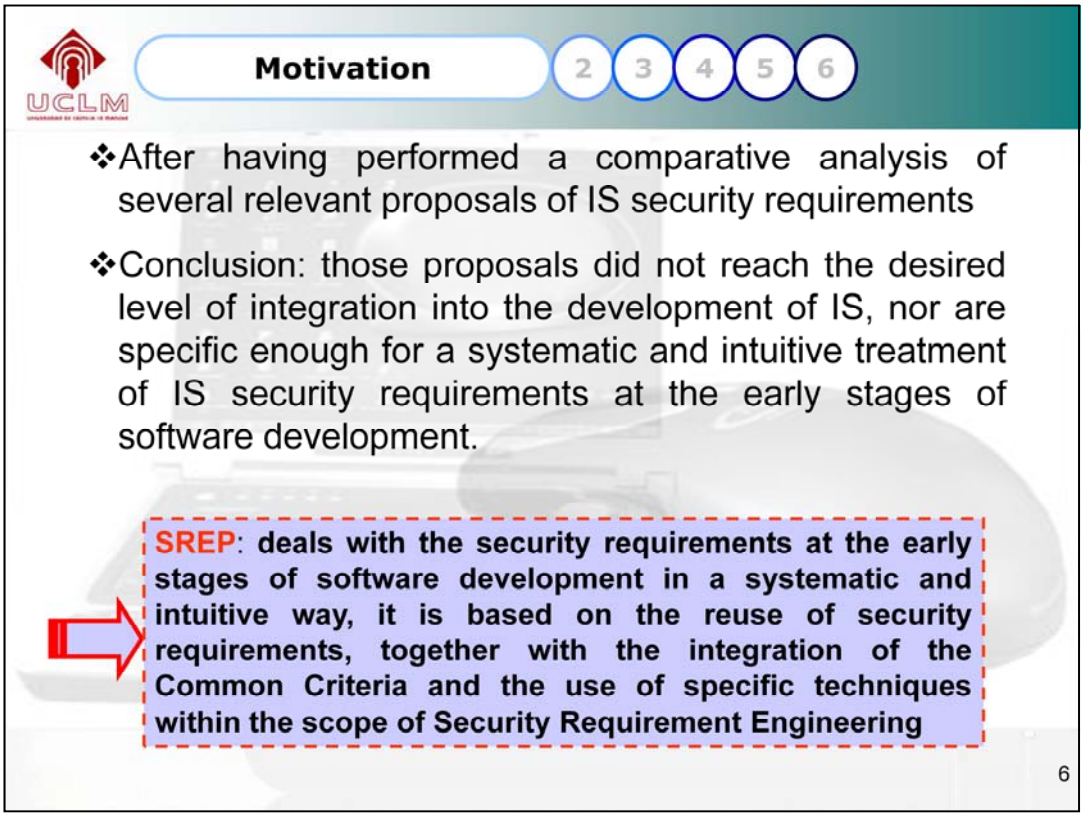
Motivation

2 3 4 5 6

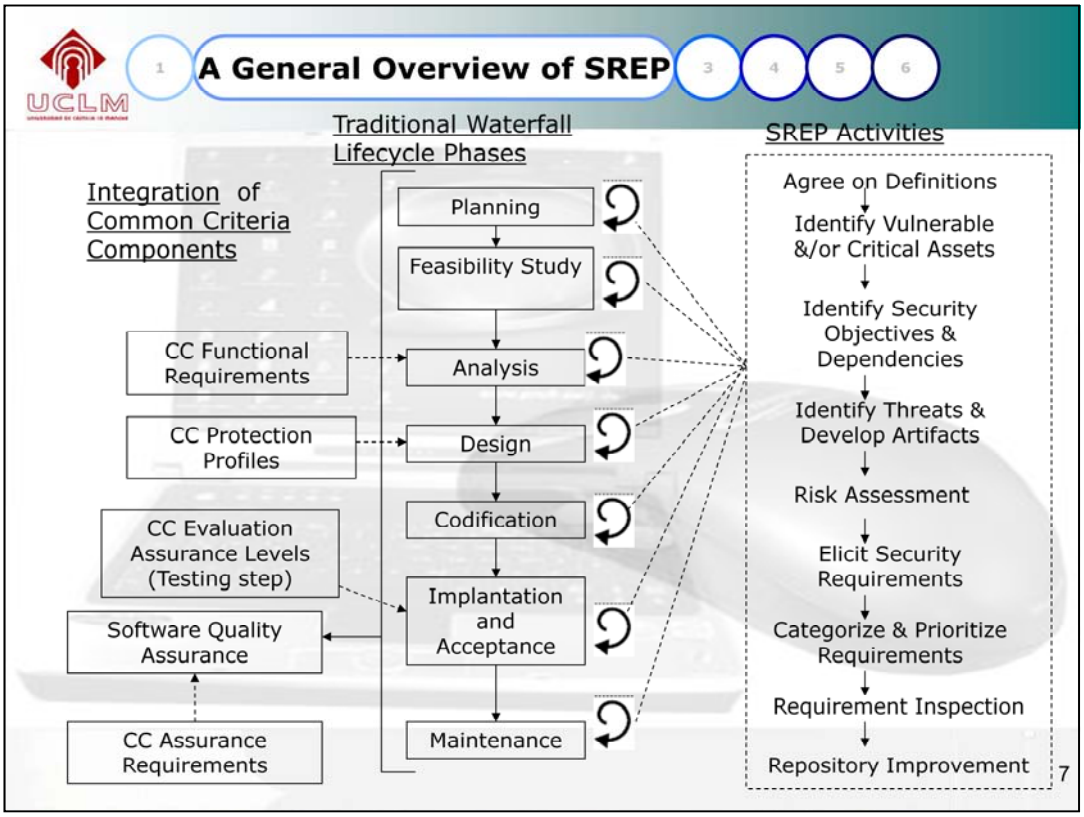
- As it is common that security requirements are undervalued and not well understood

→A very important part of the achieving of secure software systems in the software development process is that known as **Security Requirements Engineering**,

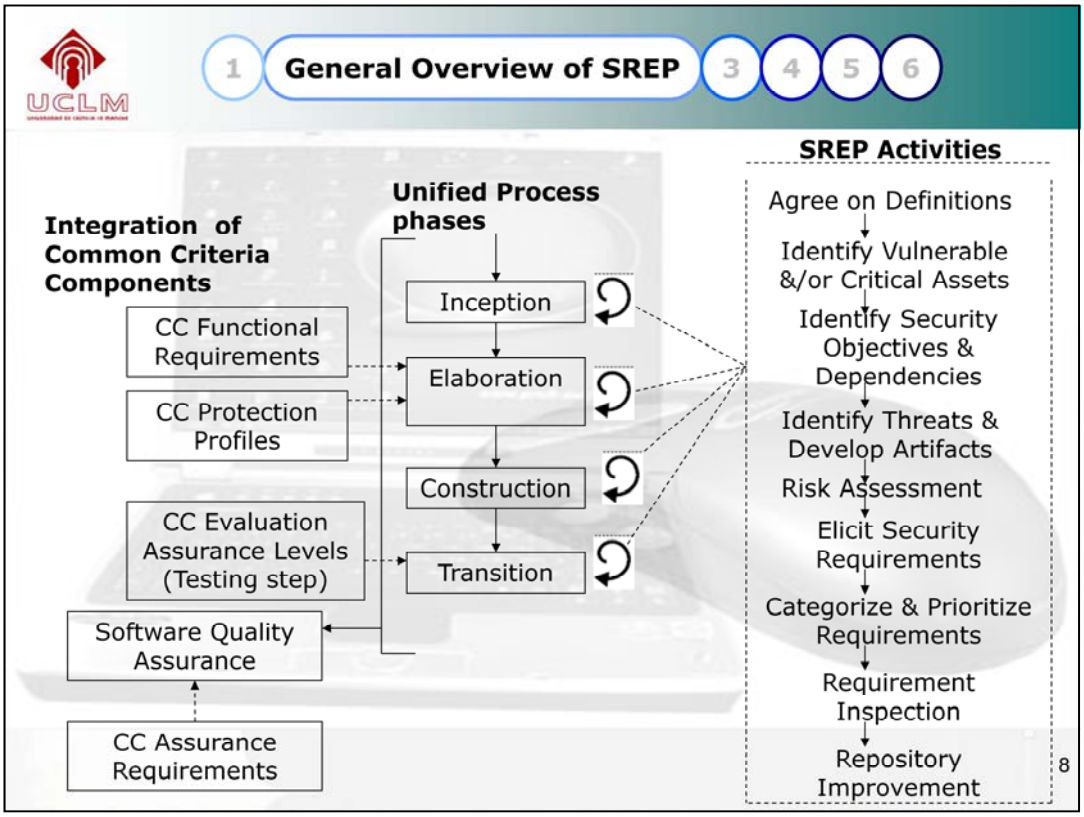
→which provides techniques, methods and norms for tackling this task in the IS development cycle



After having performed a comparative analysis of several relevant proposals of IS security requirements we concluded that those proposals did not reach the desired level of integration into the development of IS, nor are specific enough for a systematic and intuitive treatment of IS security requirements at the early stages of software development. In addition, as yet, only few works (such as the article of Massacci et al.) describes complex case studies which really cope with the complexity required by security standards. Therefore, in this paper we briefly present the Security Requirements Engineering Process (SREP) along with a case study of this proposal, which describes how to integrate security requirements into the software engineering process in a systematic and intuitive way. In order to achieve this goal, our approach is based on the integration of the Common Criteria (CC) into the software lifecycle model, because the CC helps us deal with the security requirements along all the IS development lifecycle, together with the reuse of security requirements which are compatible with the CC Framework subset. In addition this proposal integrates other approaches such as UMLSec , security use cases or misuse cases.

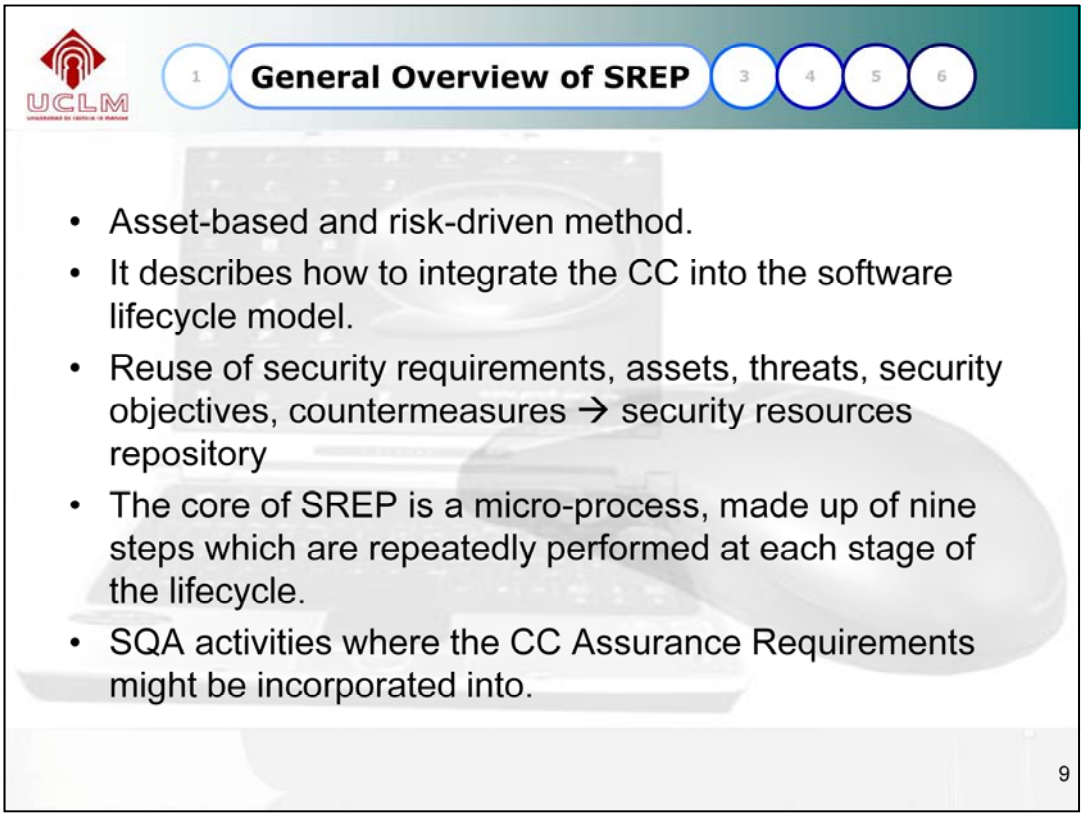


On the right we can see the 9 steps of SREP and how is integrated in traditional waterfall lifcycle phases and how the CC components are incorporated in the process.



As it is described in Fig. 1, the UP lifecycle is divided into a sequence of phases, and each phase may include many iterations. Each iteration is like a mini-project and it may contain all the core workflows (requirements, analysis, design, implementation, and test), but with different emphasis depending on where the iteration is in the lifecycle. Moreover, the core of SREP is a micro-process, made up of nine activities which are repeatedly performed at each iteration throughout the iterative and incremental development, but also with different emphasis depending on what phase of the lifecycle the iteration is in.

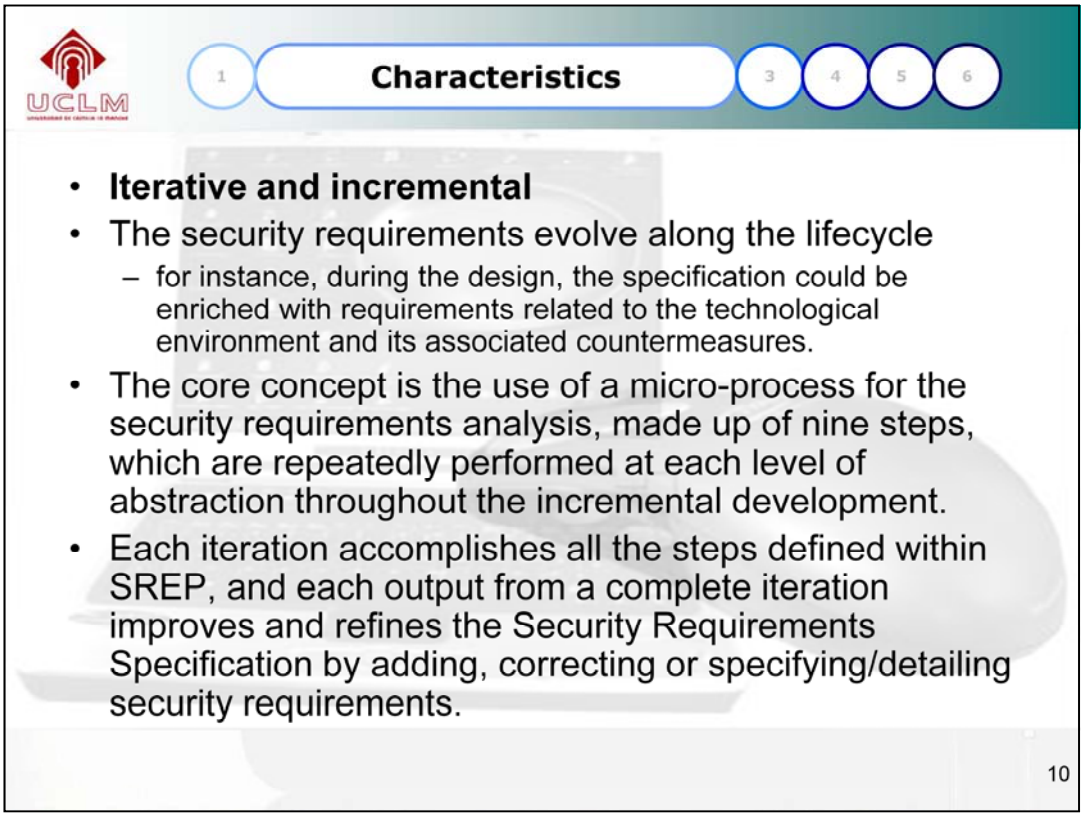
Thus, the model chosen for SREP is iterative and incremental, and the security requirements and their associated security elements (threats, security objectives, etc.) evolve along the lifecycle. At the same time, the CC Components are introduced into the software lifecycle, so that SREP uses different CC Components according to the phase of the lifecycle and the activity of SREP, although the Software Quality Assurance (SQA) activities are performed along all the phases of the software development lifecycle, and it is in these SQA activities where the most of CC Assurance Requirements might be incorporated into.



SREP is an asset-based and risk-driven method for the establishment of security requirements in the development of secure Information Systems. Basically, this process describes how to integrate the CC into the software lifecycle model together with the use of a security resources repository to support reuse of security requirements, assets, threats and countermeasures. The focus of this methodology seeks to build security concepts at the early phases of the development lifecycle.

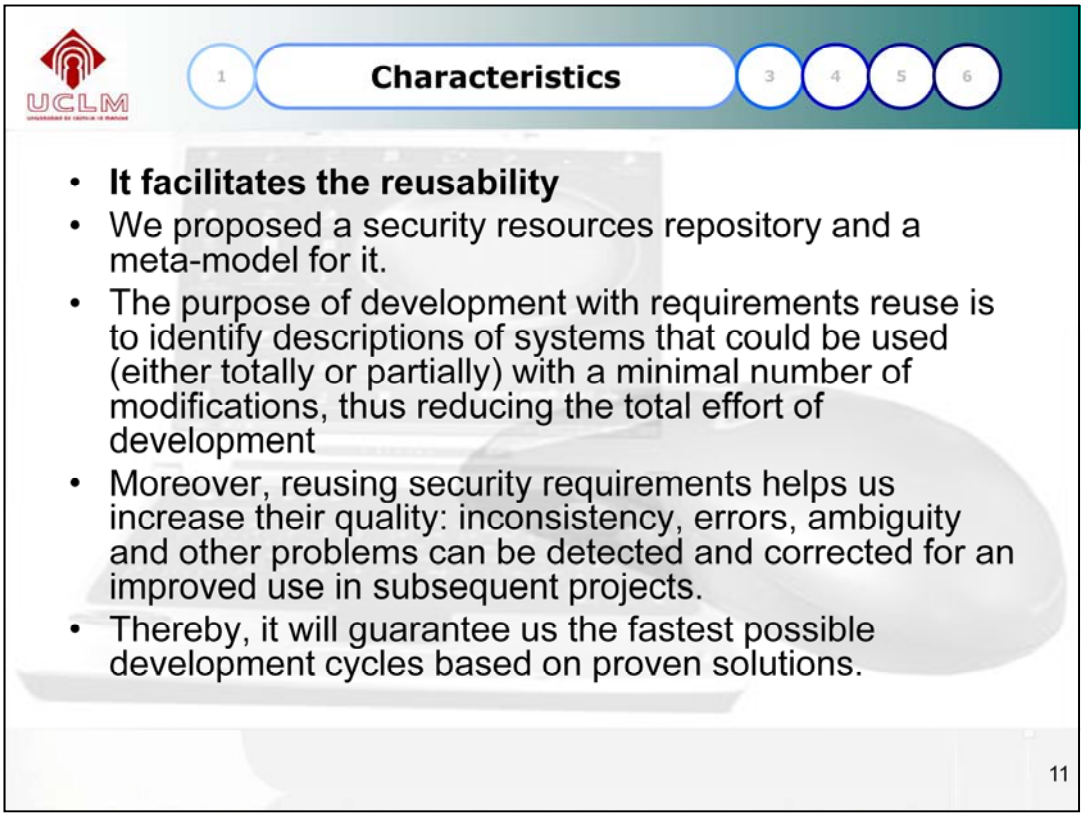
The core of SREP is a micro-process, made up of nine activities which are repeatedly performed at each iteration throughout the iterative and incremental development, but also with different emphasis depending on what phase of the lifecycle the iteration is in.

At the same time, the CC Components are introduced into the software lifecycle, so that SREP uses different CC Components according to the phase of the lifecycle and the activity of SREP, although the Software Quality Assurance (SQA) activities are performed along all the phases of the software development lifecycle, and it is in these SQA activities where the most of CC Assurance Requirements might be incorporated into.



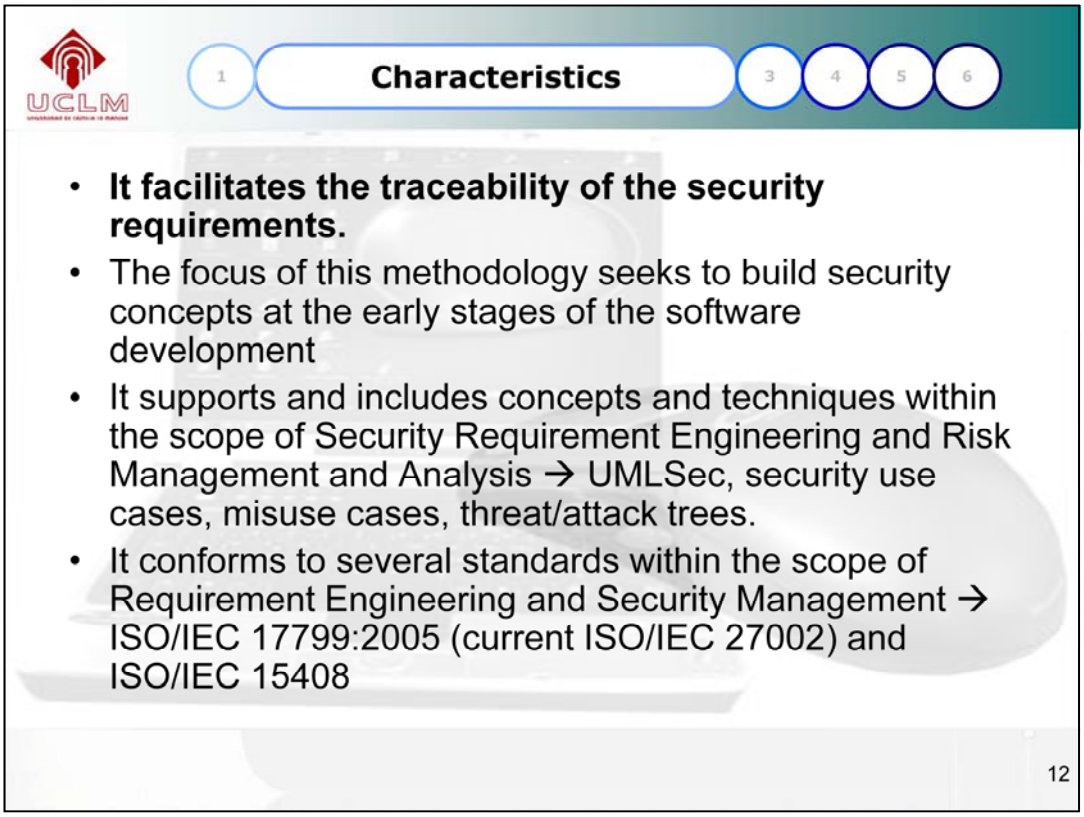
In general terms the main characteristics of SREP are:

•Iterative and incremental. The model chosen for SREP is iterative and incremental, thus the security requirements evolve along the lifecycle; for instance, during the design, the specification could be enriched with requirements related to the technological environment and its associated countermeasures. The core concept is the use of a micro-process for the security requirements analysis [2], made up of nine steps, which are repeatedly performed at each level of abstraction throughout the incremental development. Each iteration accomplishes all the steps defined within SREP, and each output from a complete iteration improves and refines the Security Requirements Specification by adding, correcting or specifying/detailing security requirements.



In general terms the main characteristics of SREP are:

•It facilitates the reusability. We proposed a security resources repository and a meta-model for it (based on Sindre, Firesmith and Opdahl approach [18]). The purpose of development with requirements reuse is to identify descriptions of systems that could be used (either totally or partially) with a minimal number of modifications, thus reducing the total effort of development [3]. Moreover, reusing security requirements helps us increase their quality: inconsistency, errors, ambiguity and other problems can be detected and corrected for an improved use in subsequent projects [19]. Thereby, it will guarantee us the fastest possible development cycles based on proven solutions.

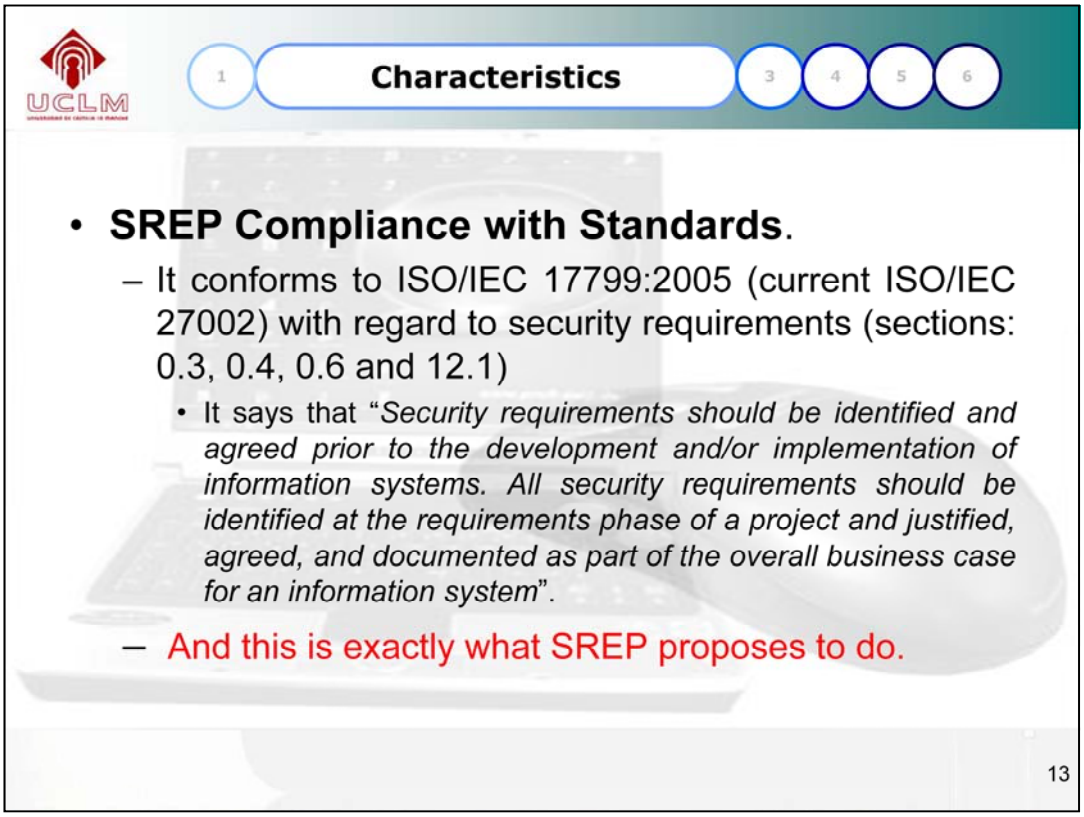


In general terms the main characteristics of SREP are:

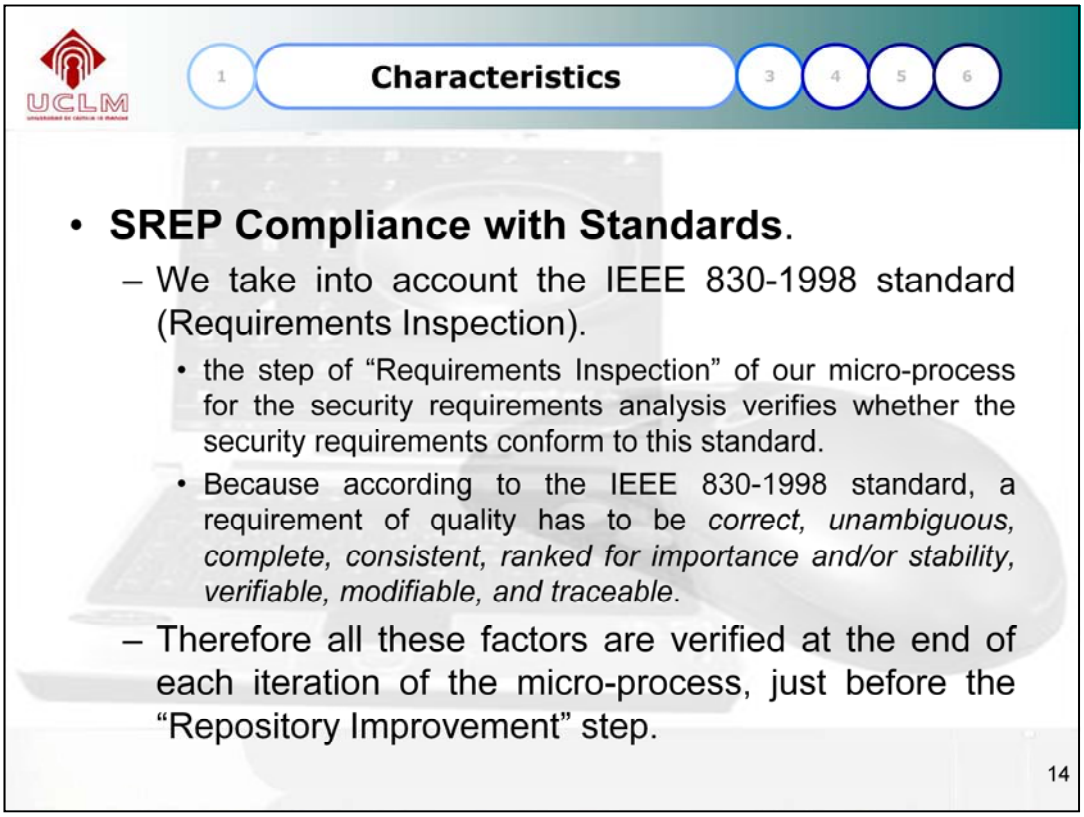
•It facilitates the traceability of the security requirements along the levels of abstraction, thanks to the structure of the repository.

•It supports and includes concepts and techniques within the scope of Security Requirement Engineering and Risk Management and Analysis, such as UMLSec, security use cases, misuse cases, threat/attack trees.

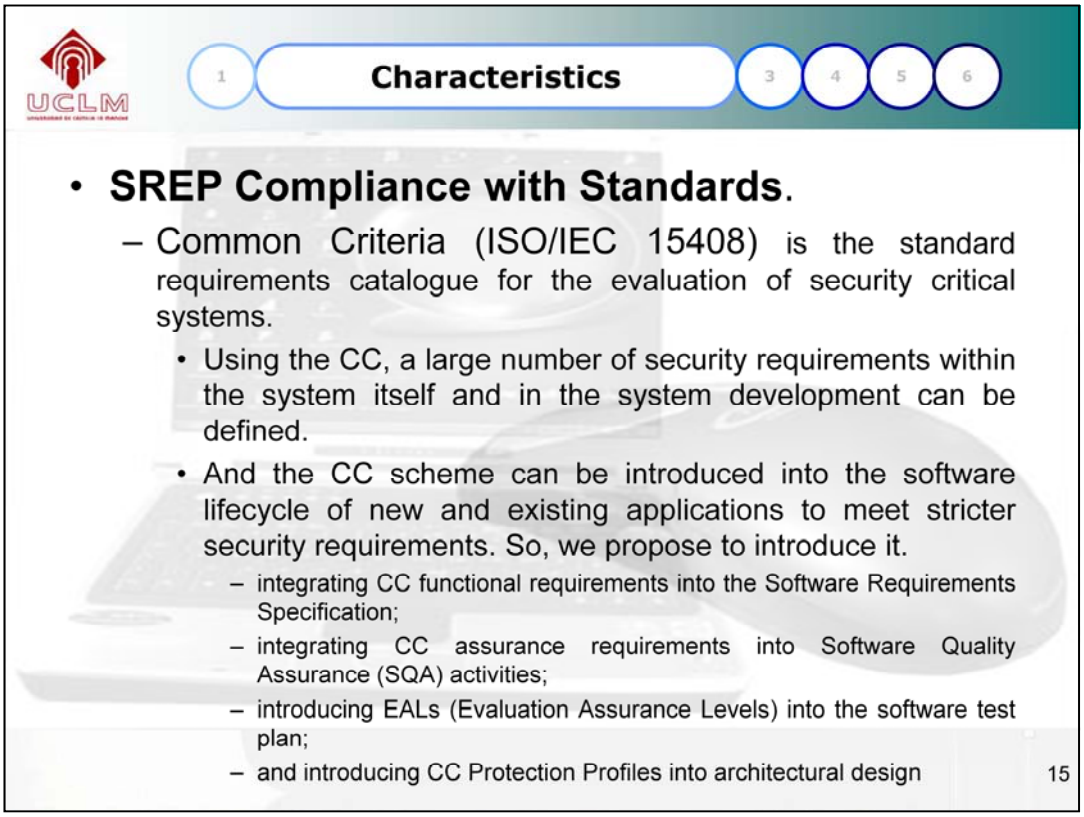
•Finally, it conforms to several standards within the scope of Requirement Engineering and Security Management, like ISO/IEC 17799:2005 and ISO/IEC 15408.



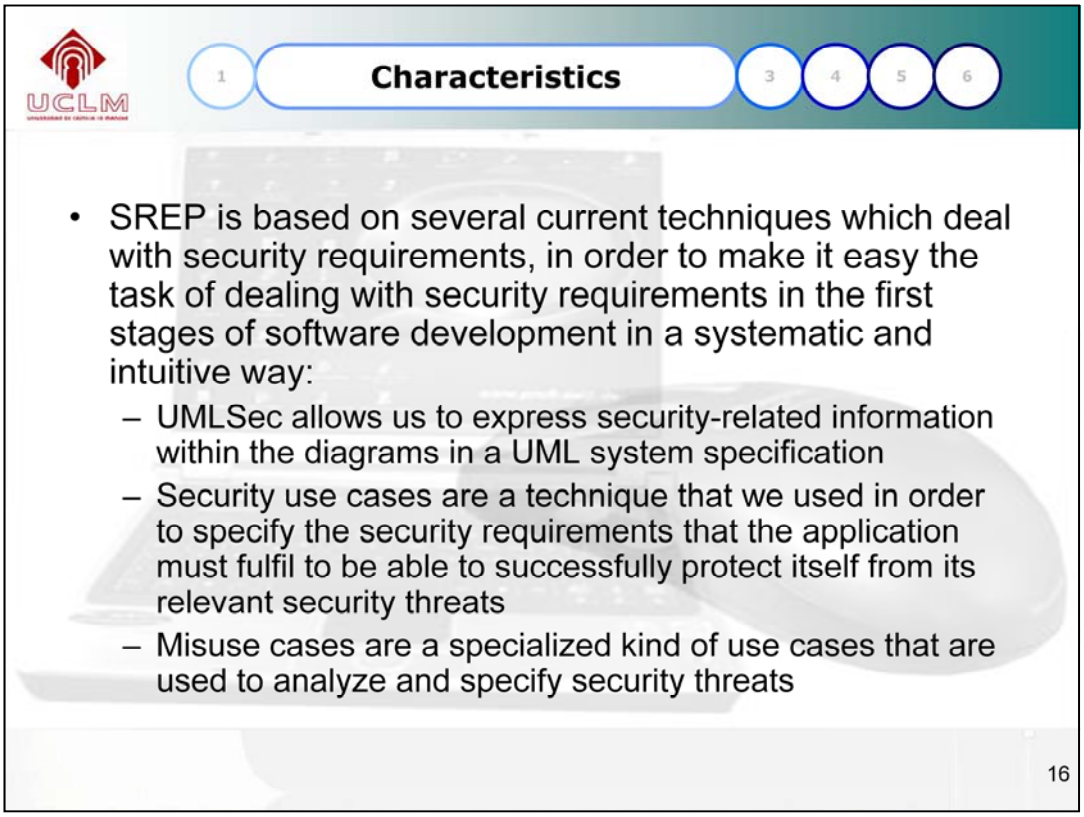
•SREP conforms to ISO/IEC 17799:2005 recommendation with regard to security requirements. It says that "Security requirements should be identified and agreed prior to the development and/or implementation of information systems. All security requirements should be identified at the requirements phase of a project and justified, agreed, and documented as part of the overall business case for an information system". And this is exactly what SREP proposes to do.



•Moreover, we take into account the IEEE 830-1998 standard, so that the step of "Requirements Inspection" of our micro-process for the security requirements analysis verifies whether the security requirements conform to this standard. Because according to the IEEE 830-1998 standard, a requirement of quality has to be correct, unambiguous, complete, consistent, ranked for importance and/or stability, verifiable, modifiable, and traceable. Therefore all these factors are verified at the end of each iteration of the micro-process, just before the "Repository Improvement" step.



The CC (ISO/IEC 15408) is the standard requirements catalogue for the evaluation of security critical systems. Using the CC, a large number of security requirements within the system itself and in the system development can be defined. And the CC scheme can be introduced into the software lifecycle of new and existing applications to meet stricter security requirements. So, we propose to introduce it. This can be accomplished, according to Kam [9], by: integrating CC functional requirements into the Software Requirements Specification; integrating CC assurance requirements into Software Quality Assurance (SQA) activities; introducing EALs (Evaluation Assurance Levels) into the software test plan; and introducing CC Protection Profiles into architectural design. Although a detailed explanation of the latter ones is outside the scope of this paper.

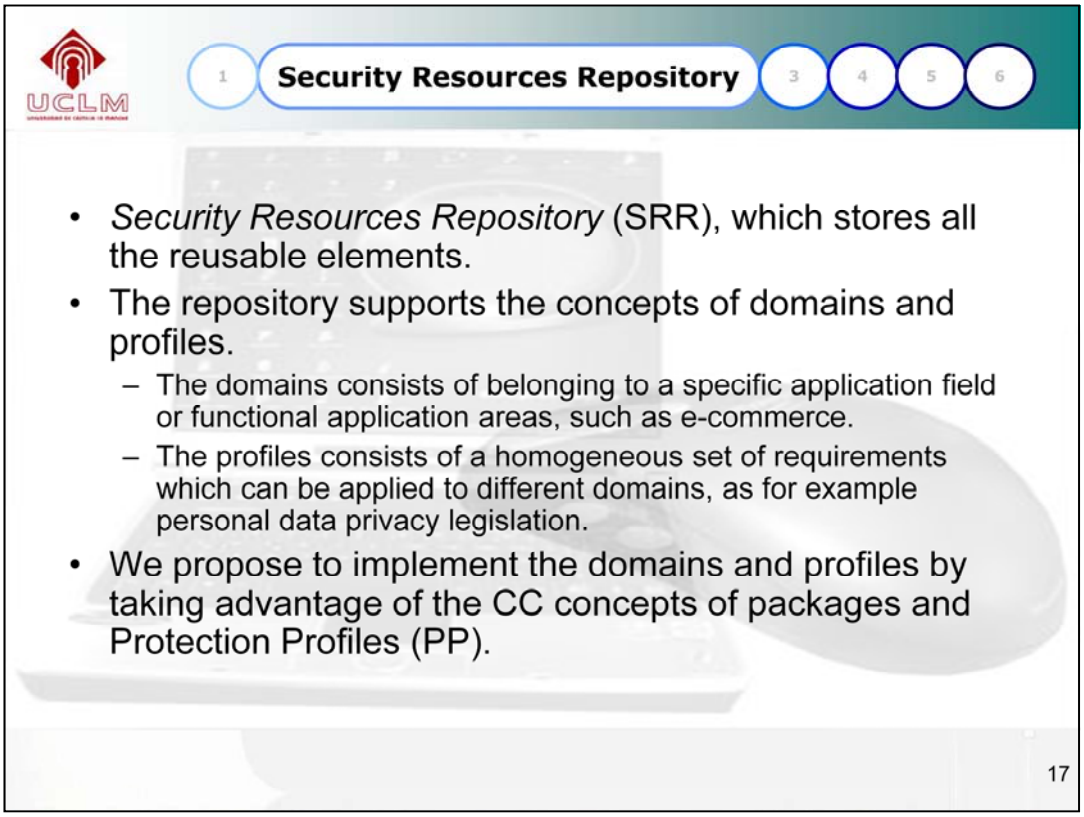


SREP is based on several current techniques, which deal with security requirements, in order to make it easy the task of dealing with security requirements in the first stages of software development in a systematic and intuitive way. The main ones are exposed below.

•*UMLSec* allows us to express security-related information within the diagrams in a UML system specification, thereby it aims to be more integrated with the artefacts produced during the development process. The extension is given in the form of a UML profile using the standard UML extension mechanisms. Stereotypes are used together with tags to formulate security requirements and assumptions on the system environment; constraints give criteria that determine whether the requirements are met by the system design [17]. We used UMLSec to specify the security requirements, and it is a complement method to security use cases.

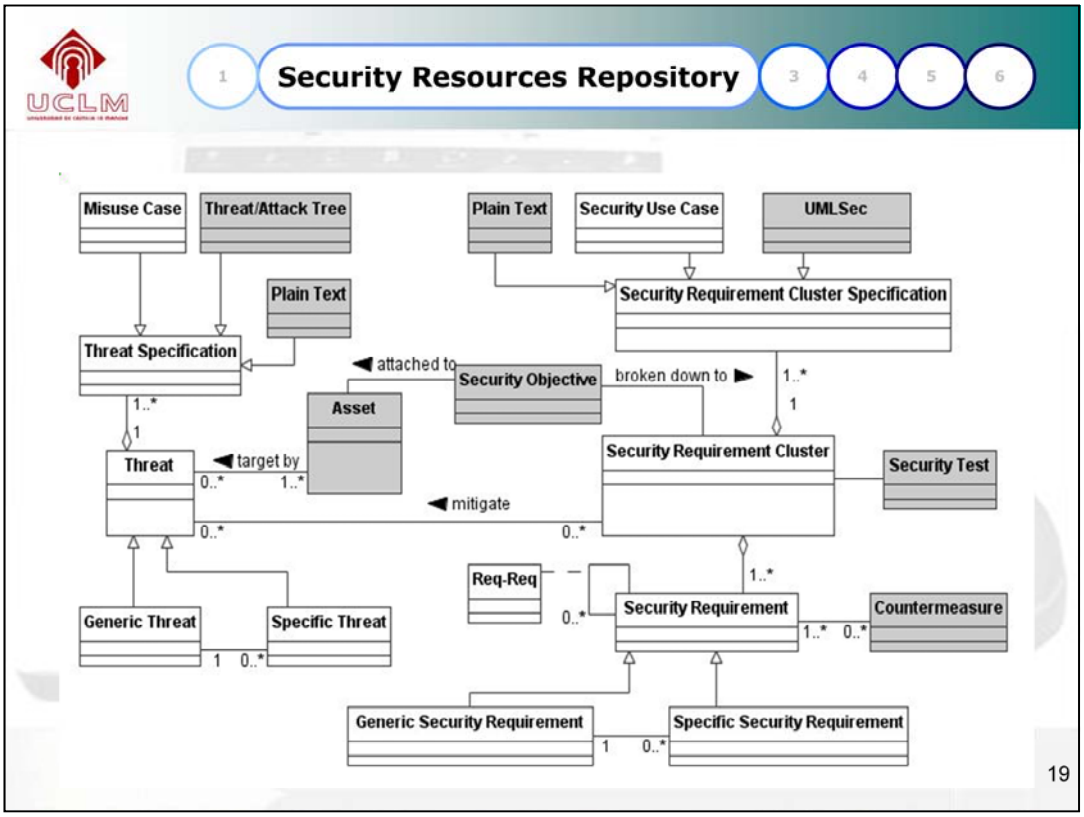
•*Security Use Cases* are a technique that we used in order to specify the security requirements that the application must fulfil to be able to successfully protect itself from its relevant security threats [6]. And they are driven by misuse cases.

•*Misuse Cases* are a specialized kind of use cases that are used to analyze and specify security threats [6]. They are the inverse of a use case, a function that the system should not allow. In more detail it might be defined as a completed sequence of actions which results in losses for the organization or some specific stakeholder [18]. In our approach they drive the security use cases, and threats are expressed as misuse cases

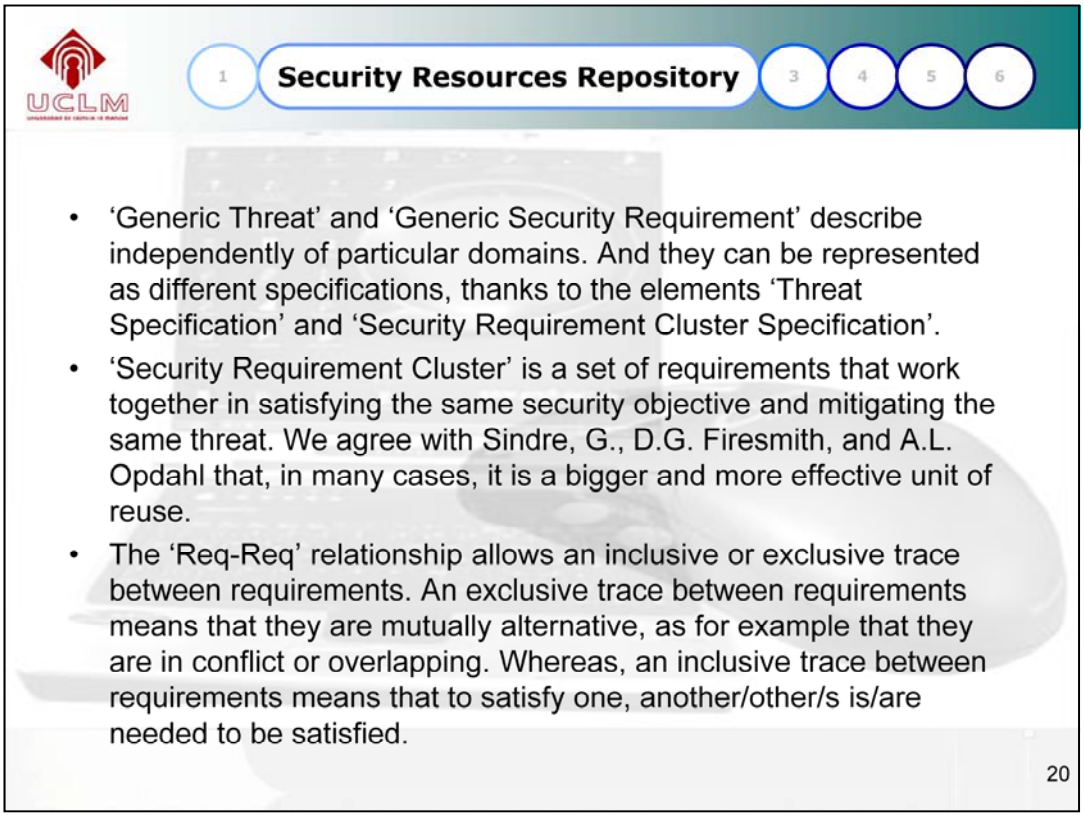


We propose a *Security Resources Repository* (SRR), which stores all the reusable elements. The repository, as SIREN [21] approach, supports the concepts of domains and profiles. The former consists of belonging to a specific application field or functional application areas, such as e-commerce. The latter consists of a homogeneous set of requirements which can be applied to different domains, as for example personal data privacy legislation. We propose to implement the domains and profiles by taking advantage of the CC concepts of packages and Protection Profiles (PP). Thus, the requirements are stored as standardized subsets of specific security requirements together with its related elements of the SRR (threats, etc.). In brief, each domain or profile is a view of the global SRR

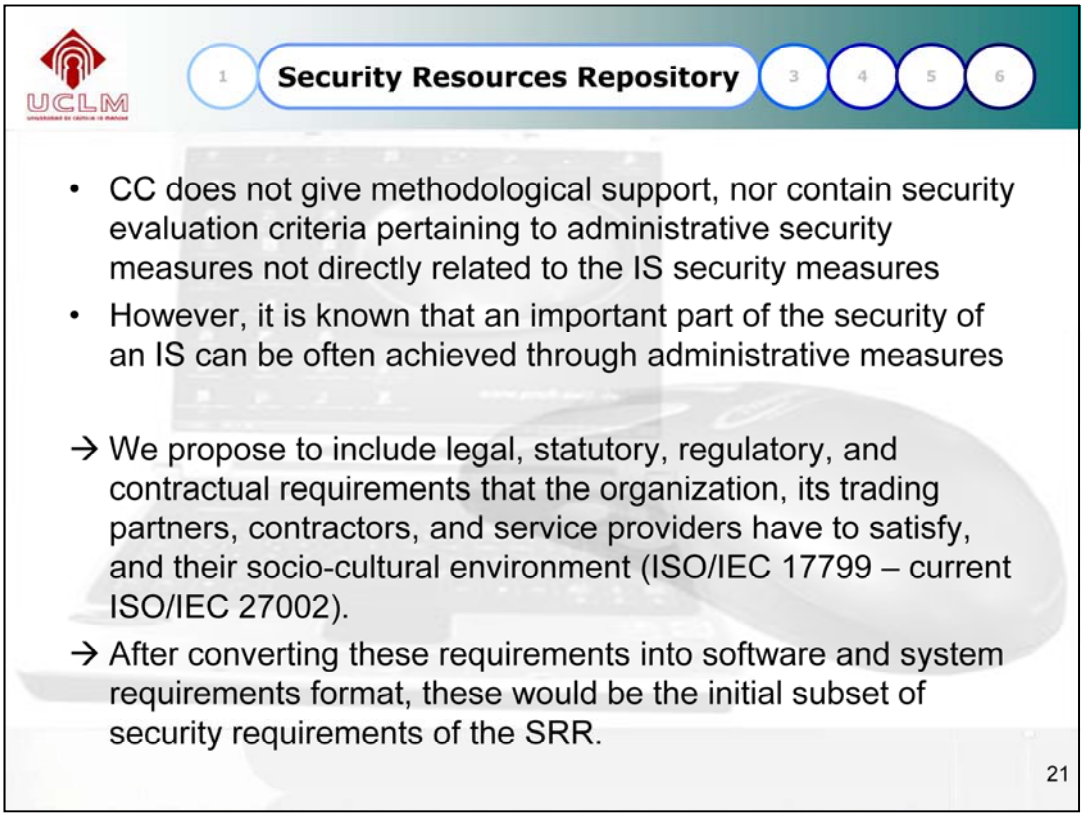
- A meta-model, which is an extension of the meta-model for repository proposed by Sindre, G., D.G. Firesmith, and A.L. Opdahl, showing the organization of the SRR is exposed in the figure.
  - The dark background in the objects represents our contribution to the meta-model



As it is presented above, it is an asset-driven as well as a threat-driven meta-model, because the requirements can be retrieved via assets or threats. Next, we will outline the most important and/or complex aspects of the meta-model



In addition, there could have been links further on to design level specifications, security test cases, countermeasures, etc. Due to the fact that our proposed model process is based on the concept of iterative software construction, as we will explain in the next section



•Finally, we would like to point out the fact that using the CC, a large number of security requirements on the system itself and on the system development can be defined. Nevertheless, the CC does not give methodological support, nor contain security evaluation criteria pertaining to administrative security measures not directly related to the IS security measures.

•However, it is known that an important part of the security of an IS can be often achieved through administrative measures.

•Therefore, according to ISO/IEC 17799:2005, we propose to include legal, statutory, regulatory, and contractual requirements that the organization, its trading partners, contractors, and service providers have to satisfy, and their socio-cultural environment.

•After converting these requirements into software and system requirements format, these would be the initial subset of security requirements of the SRR.

•Moreover, if the organization has any activity in Spain we propose that the SRR contains all the requirements taken from MAGERIT, the Spanish public administration risk analysis and management method, which conforms to ISO 15408, as well as lists of assets, threats and countermeasures. This way, it will constitute a profile which conforms to Spanish security and data privacy protection legislation

Process Model

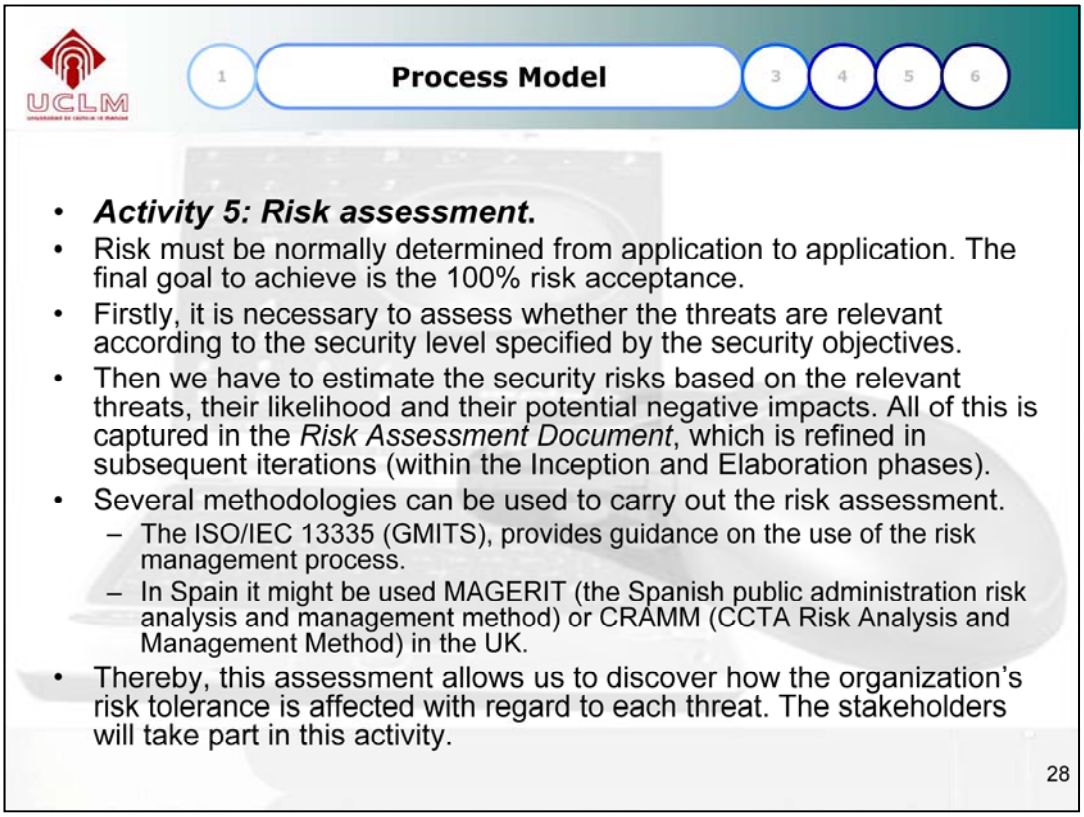
- *Step 1: Agree on definitions*
- *Step 2: Identify vulnerable and/or critical assets*
- *Step 3: Identify security objectives and dependencies*
- *Step 4: Identify threats and develop artefacts.*
- *Step 5: Risk assessment*
- *Step 6: Elicit security requirements*
- *Step 7: Categorize and prioritize requirements*
- *Step 8: Requirements inspections*
- *Step 9: Repository improvement*

22

- At the same time, as we integrate into these nine steps the CC security functional requirements, we propose to outline the EALs in the software test plan and then verify them during test execution.

- And parallely, we proposed to introduce the CC security assurance requirements into SQA activities, like quality control, defect prevention and defect removal activities. →the configuration management plan is the first activity that is explicitly required to fulfil the CC security assurance requirements

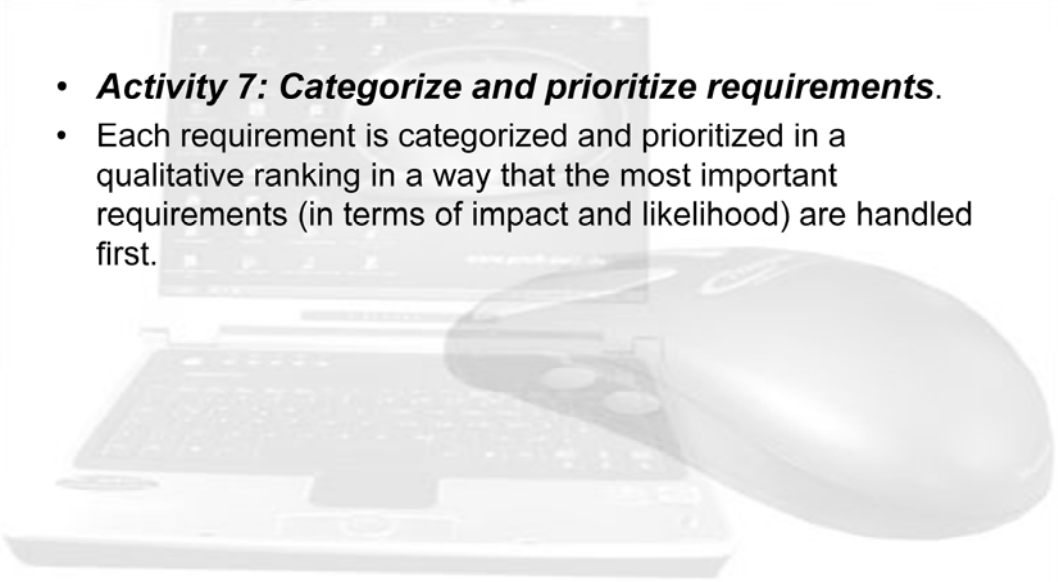
- ***Activity 1: Agree on definitions.***
- The first task for the organization is to define the stakeholders and to agree upon a common set of security definitions, along with the definition of the organizational security policies and the security vision of the IS.
- It is in this activity when the *Vision Document* artefact is created and it must contain the general vision of the IS with a special focus on security aspects.
- In addition the stakeholders will participate in these latter tasks, and the candidate definitions will be mainly taken from ISO/IEC and IEEE standards, such as ISO/IEC 13335, ISO/IEC 17799:2005 (current 27002), ISO/IEC 27001:2005, ISO/IEC 9126, IEEE Std. 830:1998, or IEEE Std. 1061-1992



We use MAGERIT

- ***Activity 6: Elicit security requirements***.
- Here, the SRR is used again. For each threat retrieved from the repository, one or more associated clusters of security requirements may be found.
- The suitable security requirements or the suitable cluster of security requirements that mitigate the threats at the necessary levels with regard to the risk assessment must be selected.
- However, additional requirements or clusters of requirements may be found by other means. Moreover, it might be specified the security test for each security requirement cluster, as well as an outline of the countermeasures for each security requirement, although they are refined at the design stage.
- Nevertheless, we agree with Firesmith in the fact that care should be taken to avoid unnecessarily and prematurely architectural mechanisms specification.
- Thus, at the end of this activity and according to ISO/IEC 17799:2005 (current ISO/IEC 27002) it must have been specified the functional, assurance, and organizational security requirements, along with the security requirements for the IT development and operational environment.
- Thereby, the *Security Requirements Specification Document* is created and refined in subsequent iterations

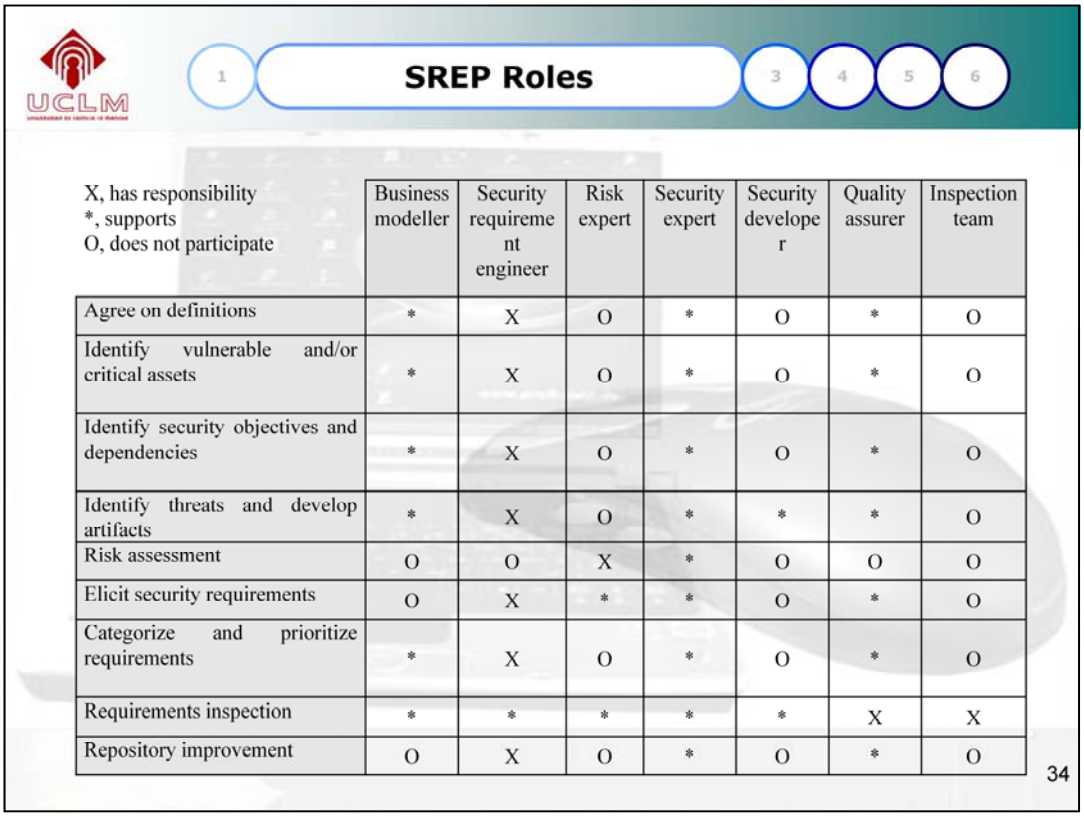
- ***Activity 7: Categorize and prioritize requirements***.
- Each requirement is categorized and prioritized in a qualitative ranking in a way that the most important requirements (in terms of impact and likelihood) are handled first.



- ***Activity 8: Requirements inspection.***
- Requirements inspection is carried out in order to validate all the generated artifacts (all the documents, requirements, the modified model elements and the new generated model elements) and it is generated a *Validation Report*.
- Its aim is to review the quality of the team's work and deliverables as well as assesses the security requirements engineering process. So, it is used as a sanity check.
- Moreover, it is verified whether the security requirements conform to the IEEE 830-1998 standard, because according to this standard, a requirement of quality has to be correct, unambiguous, complete, consistent, ranked for importance and/or stability, verifiable, modifiable, and traceable.
- After all, the security requirements documentation is written, so that a S*ecurity Requirements Rationale Document* is provided, showing that if all the security organizational, functional and assurance requirements are satisfied and all security objectives are achieved, the defined security problem is solved:
  - all the threats are countered, the organizational security policies are enforced and all assumptions are upheld.

- ***Activity 8: Requirements inspection.***
- Furthermore, it is performed within the Test workflow of the UP and with the help of the CC assurance requirements and EALs (Evaluation Assurance Level) and the SSE-CMM (ISO/IEC 21827).
- Thereby, we propose to evaluate the security of the IS along with the security engineering process by using the CC assurance requirements and the SSE-CMM at the same time with the help of CC_SSE-CMM .
- Thus referring to CC_SSE-CMM Part 3, the Process Area (PA) in association with CC EAL can be selected and based on the PA selected it can be determined the current level of SSE-CMM operation capability and extract the path for the better operation capability level .
- Thus, it can be assured that a security IS with a high reliability will be developed by conducting the CC evaluation and the SSE-CMM evaluation at the same time.
- Additionally, this activity is carried out by the quality assurer and by the inspection team at the last phase (Transition phase), with the participation of the stakeholders and security requirements engineers mainly

- ***Activity 9: Repository improvement***.
- The new model elements (threats, requirements, etc…) found throughout the development of the previous activities and which are considered as likely to be used in forthcoming applications and with enough quality, according to the Validation Report, are introduced into the SRR.
- Furthermore, the model elements already in the repository could be modified in order to improve their quality.
- Thereby, all these new or modified model elements / artifacts, which have been introduced into the SRR, altogether constitute a baseline.
- After that the *Security Target* or *Protection Profile documents* of the CC are written.
- This activity will be performed coinciding with the milestone at the end of each phase of the UP.



The roles defined here constitute a supplement to the roles in software engineering, the difference is that these roles are especially focused on security and also require special training

SREP Roles

→ These roles are a supplement to the roles in software engineering, but are especially focused on security and also require special training

- ***Business modeller***.
  - He/she describes the business processes, the roles involved and the artifacts produced or used in the process.
  - He/she helps develop artifacts in SREP (like misuse cases, etc.) and construct the processes in a security-enhanced way, which fit in the business model of the IS.

SREP Roles

- ***Security requirement engineer***.
- This is the key role and it participates and leads most activities.
- It is in charge of the security vision of the IS, it also identifies the assets, the security objectives and its dependencies and the threats, and elicits and specifies the requirements, as well as categorizes and prioritizes the requirements with the help of other kind of specialists (if needed).
- Depending on the size of the project more than one person can be assigned to this role.
- Furthermore, this role must not necessarily have a thorough technical understanding of security, although a sound security management is required.

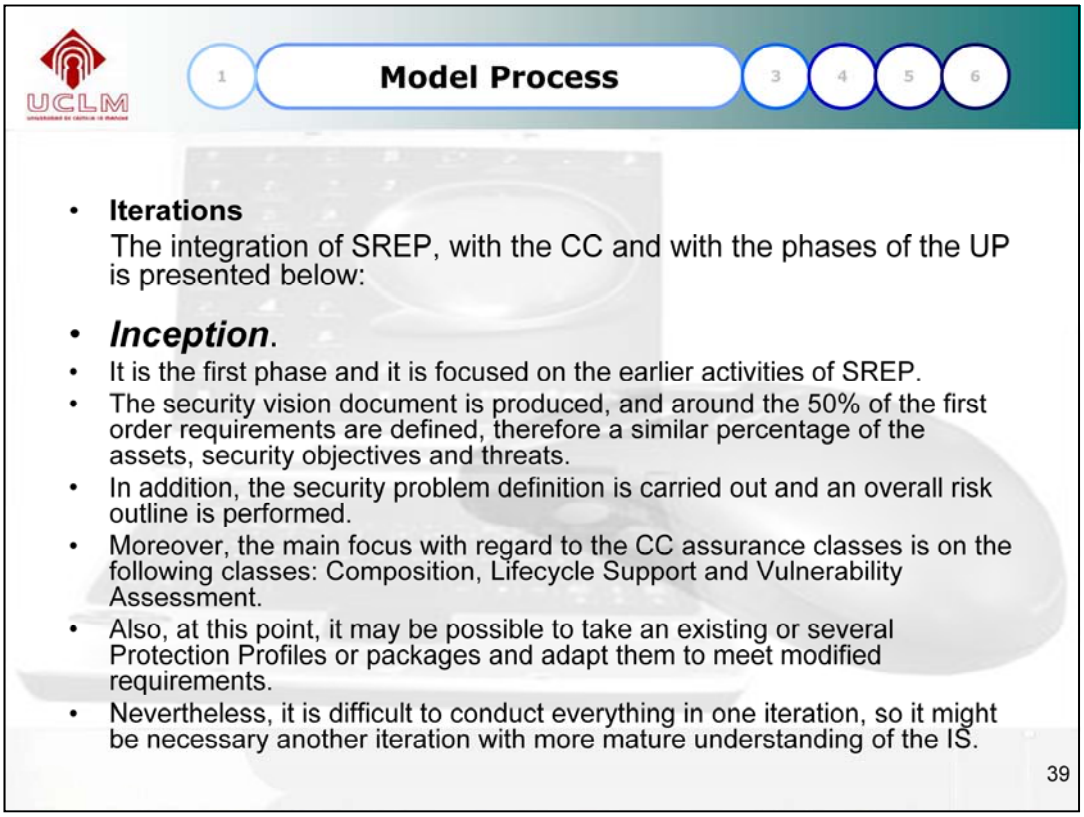
SREP Roles

- ***Security expert***.
- The main task of the security expert is to improve the overall security of the IS.
- This role is the technical expert in security so that he/she acts as a consultant, and helps us find security relevant information, estimate the degree to which IS meets its security claims and define the security vision of the IS and the organizational security policies and measures.
- ***Security developer***.
- The role of the security developer is to support the construction of tests to help the Requirements Inspection activity during the Test workflow of the UP.

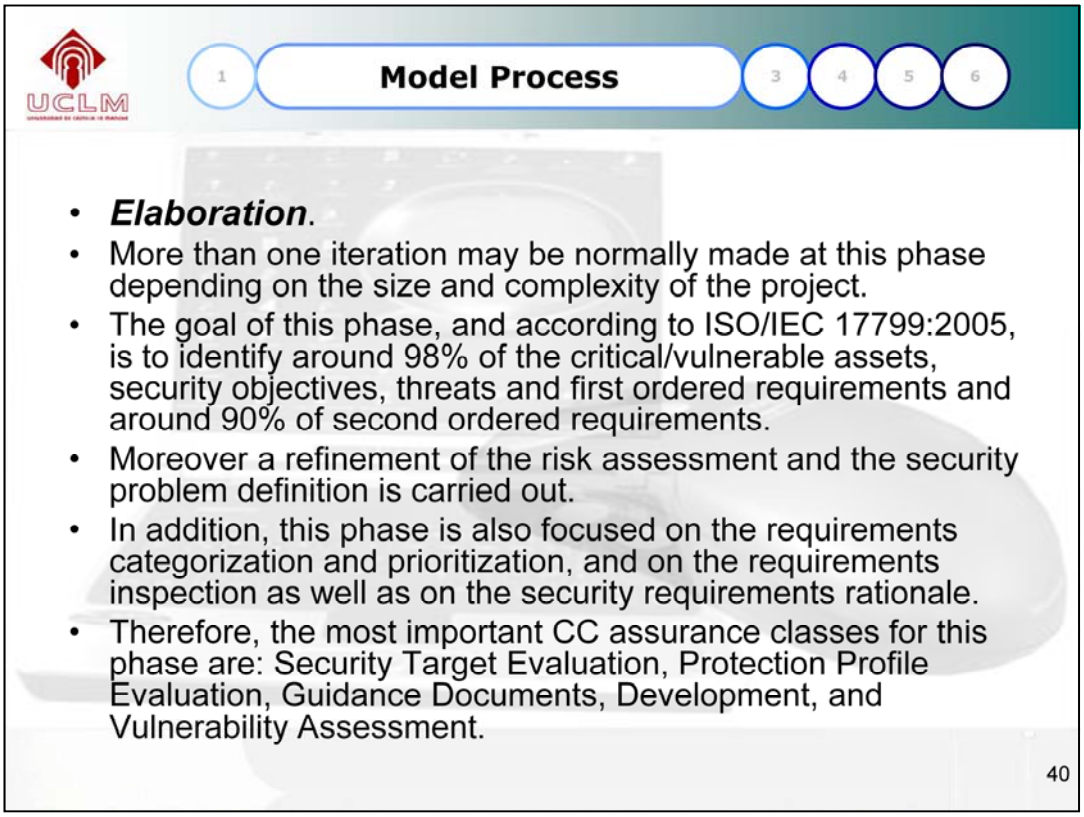
SREP Roles

- ***Quality assurer***.
- This is the role responsible for the Requirements Inspection activity within the Test workflow of the UP and it could take advantage of the use of the CC assurance classes.
- In addition, this role can help us with informal reviews of the quality of the most important artifacts in each activity.
- ***Inspection team***.
- It is a group external to the IS development team whose aim is to review the quality of the development team's work and deliverables as well as evaluate the security engineering process by using the CC assurance requirements and the SSE-CMM, with the help of CC_SSE-CMM [12].
- Besides it is the role responsible for the Requirements Inspection activity within the Transition phase of the UP. Additionally, this team is in charge of the assurance that the IS meets its security claims with the help of the EALs.

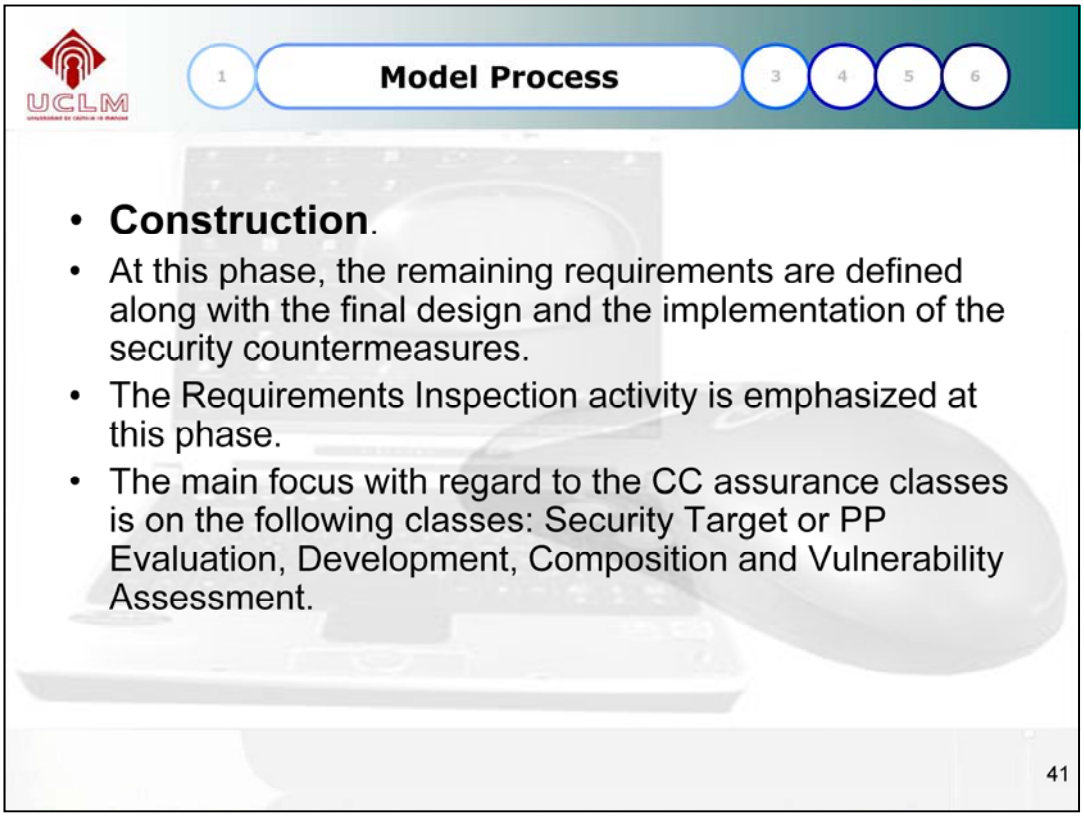
38



We propose an iterative and incremental security requirements engineering process, so that each iteration coincides with an iteration within a phase of the UP. This is because the UP lifecycle is divided into a sequence of phases, which may include many iterations, and each one concludes with a major milestone. This philosophy lets us take into account changing requirements, facilitates reuse and correct errors over several iterations, risks are discovered and mitigated earlier, and the process itself can be improved and refined along the way. Therefore, the result is a more robust IS



We propose an iterative and incremental security requirements engineering process, so that each iteration coincides with an iteration within a phase of the UP. This is because the UP lifecycle is divided into a sequence of phases, which may include many iterations, and each one concludes with a major milestone. This philosophy lets us take into account changing requirements, facilitates reuse and correct errors over several iterations, risks are discovered and mitigated earlier, and the process itself can be improved and refined along the way. Therefore, the result is a more robust IS



We propose an iterative and incremental security requirements engineering process, so that each iteration coincides with an iteration within a phase of the UP. This is because the UP lifecycle is divided into a sequence of phases, which may include many iterations, and each one concludes with a major milestone. This philosophy lets us take into account changing requirements, facilitates reuse and correct errors over several iterations, risks are discovered and mitigated earlier, and the process itself can be improved and refined along the way. Therefore, the result is a more robust IS