

# Software Requirement Patterns: A Tutorial

*Xavier Franch*

7<sup>th</sup> IEEE RCIS, May 2013  
Paris, France



Group of Software Engineering for Information Systems  
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Outline

- Background: requirements engineering
- Background: patterns in software engineering
- Software requirement patterns
- Classification schemas
- Construction process. Application process
- An example: the CRPHIT case
- Conclusions

# Preliminaries

- If you are planning to attend this tutorial, please consider to fill the survey at:  
<http://www.upc.edu/gessi/PABRE/REFSQ2013-Survey.html>
  - Will help me getting a clearer view of attendees
- If you have attended this tutorial, please consider to fill the survey at:  
<http://www.upc.edu/gessi/PABRE/REFSQ2013-Survey.html>
  - Will help me knowing more about the community position on the topic

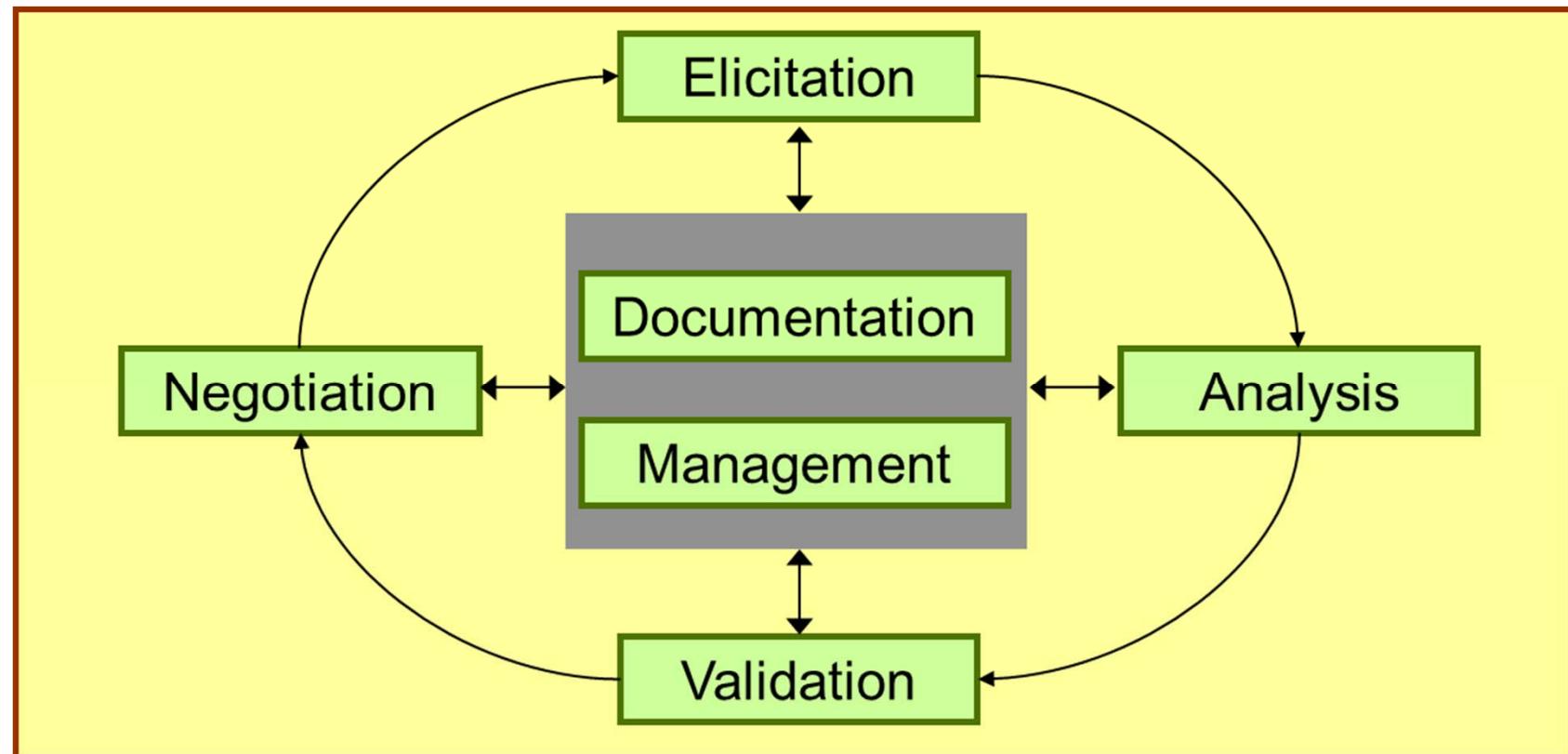
# Background



Group of Software Engineering for Information Systems  
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Requirements engineering (RE)

Set of activities around requirements



# Software requirement (1)

1. A condition or capability needed by a user to solve a problem or achieve an objective
2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document
3. A documented representation of a condition or capability as in 1 or 2

*IEEE Standard Glossary of Software Engineering Technology (1997)*

# Software requirement (2)

- Something that the product must do or a quality that the product must possess (Robertson & Robertson)
- Descriptions of how the system must behave, or of a property or attribute of the system (Sommerville & Sawyer)
- Everything that determines a design decision (Lawrence)

# Types of requirements

- Functional requirements: defines a function of a software system or its component
- Non-functional requirement: every requirement that is not functional ☺
  - Or, a requirement that defines criteria that can be used to judge the operation of a system
  - Also: quality requirement, *-ilities*, ...
  - Always controversial (Glinz. On Non-Functional Requirements. RE'07)
- Non-technical requirement: defines some condition over the system as a commercial good

# Requirements in context

- Requirement: measurable condition
  - “The system shall provide an authentication process based on voice recognition”
- Goal: high-level strategic condition
  - “The authentication process shall be secure”
- Constraint: non-negotiable situation that may affect the system
  - “There are no fingerprint-based recognition devices”
- Assumption: belief given for granted
  - “The system users will consider that voice recognition will not violate their privacy”

# Requirements specification

Software requirements specification (SRS)

- Document that compiles all the requirements
  - E.g., IEEE 830 template

# What's a pattern?

“Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice”

*Christopher Alexander*

*The Timeless Way of Building, 1977*

Job: Architect (of buildings)

# From buildings to software

“People liked our book very much. We were surprised though, when we found out computer programmers liked it, because it was about building not programming. But the programmers said, "this is great, it helps think about patterns in programming and how to write reusable code that we can call upon when we need it." ”

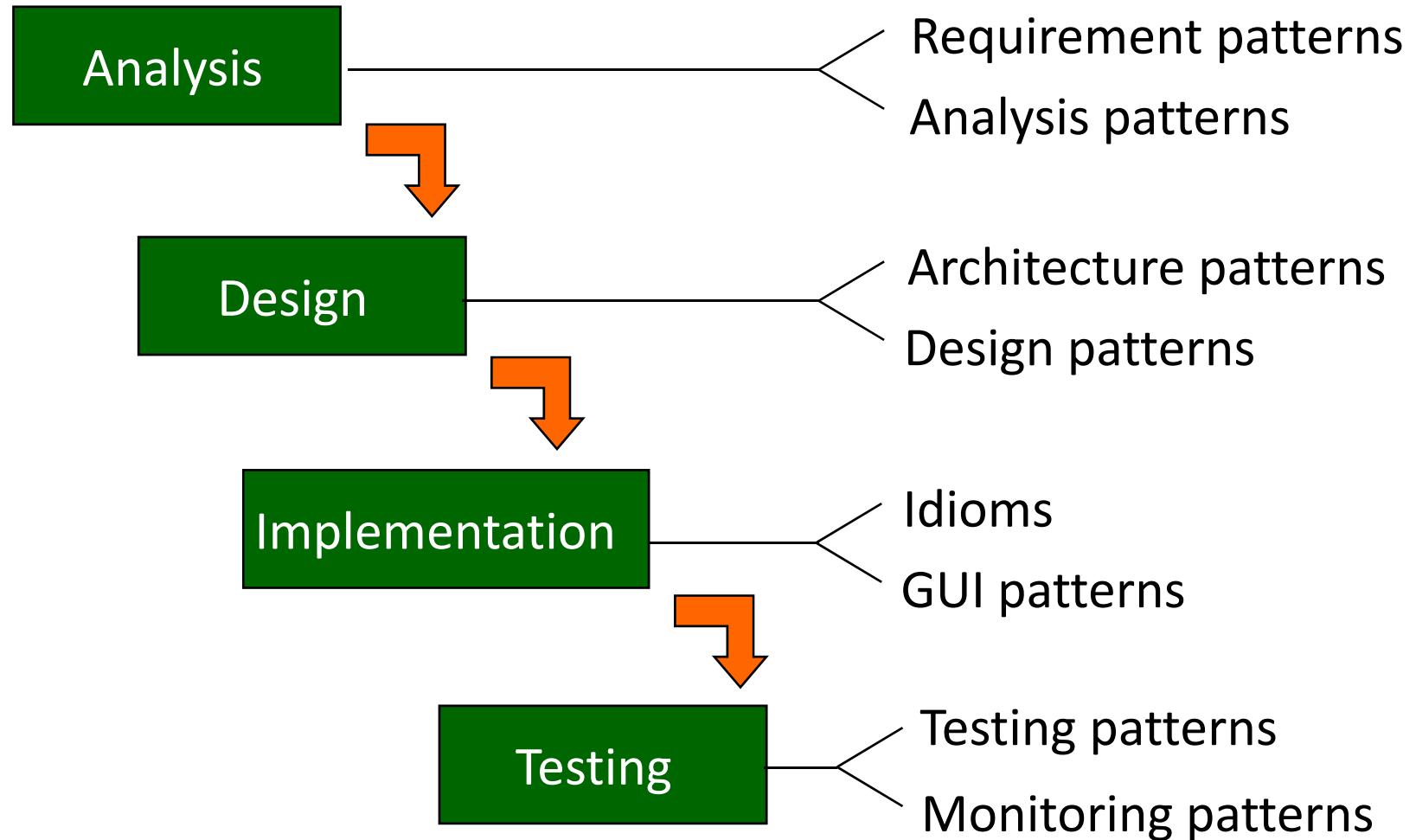
[www.patternlanguage.com](http://www.patternlanguage.com)

# In the beginning...

Design patterns:

- Arguably the most popular type of patterns
- Initial proposal:
  - Classification: creator, structural, behavioural
  - Conceptual tools: delegation, polymorphism, aggregation, ...

# But there are more...



# Software Requirement Patterns



Group of Software Engineering for Information Systems  
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Software requirement patterns (SRP)

Fundamental principle: when specifying a system, a high proportion of requirements are recurrent and belong to a relatively small number of types

- Specially in the case of non-functional requirements

Requirement pattern: an approximation to the specification of a particular type of requirement

- Generates one or more requirements

# Benefits of patterns



Guidance



Consistency



Performance

## Specially when...

- The number of software projects conducted is high
- SRSs have a high number of requirements
- The context is similar
  - Similar customers, similar projects
- The team of analysts and the team of developers is clearly differentiated
  - Communication among them needs to be precise
- The organization wants to follow prescriptive processes
  - Eventually aiming at some kind of certification

# Impact in RE

- **Elicitation**
  - More chances to discover needed requirements
- **Analysis**
  - Clearer relationships
- **Validations**
  - Systematic
- **Negotiation**
  - Consequences known right away
- **Documentation**
  - Uniform, organized, comprehensive
- **Management**
  - Traceability; changes with clearer effects

# Strategies

The use of the catalogue may be:

- Opportunistic
  - Just another elicitation tool
- Systematic
  - Strategic decision
  - Drives the elicitation process
  - Upfront investment: analysis of as many SRS as possible
  - Continuous update

# Systematic use of an SRP catalogue

- I manage the patterns catalogue.
- I decide which new patterns to add.
- I analyze the statistics of use of patterns, and use them to evolve the catalogue.
- I manage the classification schemas that may be used for browsing the catalogue.

- I manage projects.
- I generate the requirements specification document or the call for tenders document.

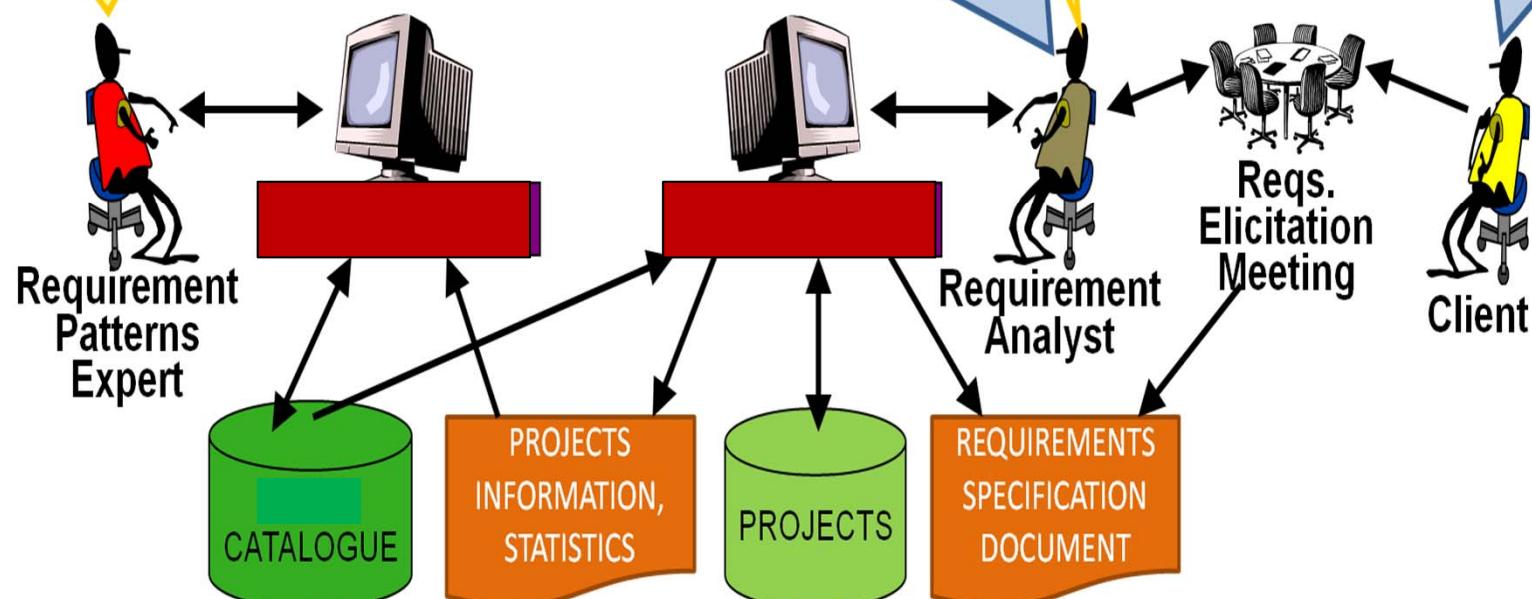
During the Reqs. Elicitation Meeting:

- I browse the patterns catalogue in order to elicit the requirements in a project.
- I look up the goal of a pattern to discuss with the client if it is relevant in their project.

- I have to fulfill my needs.

During the Reqs. Elicitation Meeting:

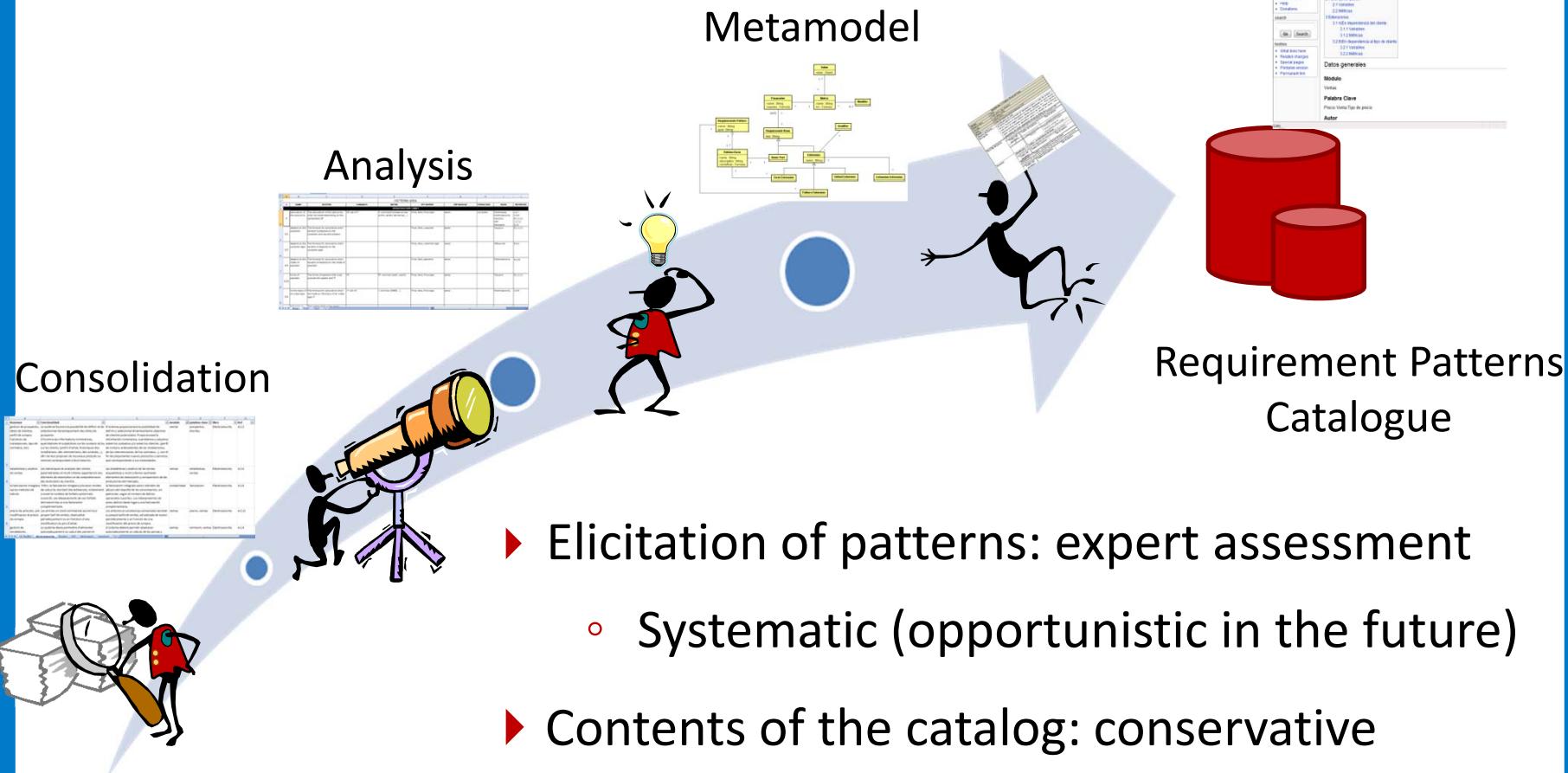
- I agree with the Requirement Analyst on whether a pattern covers some of my needs.



# Structure of an SRP

- Totally configurable
  - Not one-size-fits-all
- Here we focus in the most essential aspects
- Useful complementary artifacts
  - Glossary
  - Compilation of references
  - ...

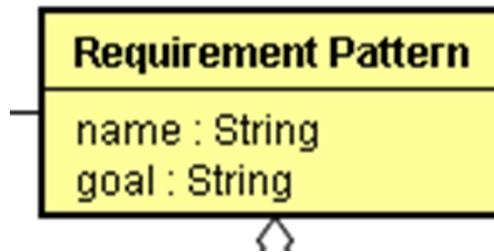
# Process



- ▶ Elicitation of patterns: expert assessment
  - Systematic (opportunistic in the future)
- ▶ Contents of the catalog: conservative
  - Style used by consultants is adopted
- ▶ Iterative
  - Driven by case studies

# The Pattern

- An approach to specifying a particular type of reqt.
  - Which is the purpose of the pattern  $\Rightarrow$  goal
  - Level of abstraction:
    - not much (oriented to writing!)
    - higher abstraction left to classification schemas



# Example

The solution should permit the trace of the modifications and access to the encrypted data in the database. The essential data to trace are: user name, date, accessed or modified data.

The solution should permit to trace all the user actions. The data to trace are: user name, date, accessed or modified data.

All the operations made over a type of object (dossier, document files, etc.) may be traced

The solution should permit to trace all the access to protected data. The data to trace are: user name, date, accessed or modified data. An historical record of this accesses must be maintained

## Pattern: Logs

**Goal:** Have data ready for audit

# Pattern forms

- A pattern may take different forms
  - Usually, mutually exclusive in a project
- Example: Data Base pattern
  - Goal: Require a particular data base technology
  - Form 1: Refer to a particular family (e.g., relational)
  - Form 2: Refer to a particular product
  - Form 3: Refer to the current system

The system shall work with a relational DB; ACME is preferred

# Basic part

- A form is characterized by a basic part
  - It captures the most basic essence of the requirement
  - Usually abstract, translates the pattern goal into a customer-oriented text -> provides structure to the doc.
- Example: *Recovering* pattern
  - **Goal:** Ensure that the system recovers after crashes
  - **Form 1:** Focused on the “what” (data)  
  
The system shall not lose data in case of crash
  - **Form 2:** Focused on the “how” (procedure)  
  
The system shall follow some recovery procedure in case of crash

# Extensions

- A form may take different shapes
  - Several factors may modify the basic behaviour
  - Extension: part of a pattern that embraces a potential requirement subordinated to other
- Example: *Failure Alerts* pattern
  - **Form:** *Homogeneous Failure Alerts*  
**Basic Part:** The system shall give an alert in case of failure  
**Extensions:**  
*Alert Types:* Lists the types of alerts provided by the system  
*Failure Types:* Lists the types of failures provided by the system

# Parameters

- Behaviour of patterns depends on certain values
  - Parameters, declared of a particular metric, may take these values
  - Parameters and/or metrics may have to fulfill correctness conditions
- Example: *Failure Alerts* pattern
  - **Form:** Homogeneous Failure Alerts
    - Basic Part: The system shall give an alert in case of failure
  - **Extensions:**
    - Alert Types: Alerts provided by the system shall be AlertTypes
    - Failure Types: Failures provided by the system shall be FailureTypes

# Applicability conditions of extensions

- They express the rules to follow for applying dependencies in a pattern
  - cardinality + dependencies among them
- Example:
  - Form: Homogeneous Failure Alerts
    - Basic Part: The system shall give an alert in case of failure
    - Alert Types: Alerts provided by the system shall be *AlertTypes*
    - Failure Types: Failures provided by the system shall be *FailureTypes*

**Applicability cond.:** Basic Form + Alert Types + Failure Types

# Applicability conditions of extensions

- They express the rules to follow for applying dependencies in a pattern
  - cardinality + dependencies among them
- Example: Online Help (OH) pattern
  - **Form:** OH Basic Features
    - Basic Part:** The system shall provide help whilst is in operation
    - Local OH:** OH shall be provided for *Functionalities*
    - Full OH:** OH shall be provided for every displayed screen
    - OH Technology:** OH shall be built using *Technologies*

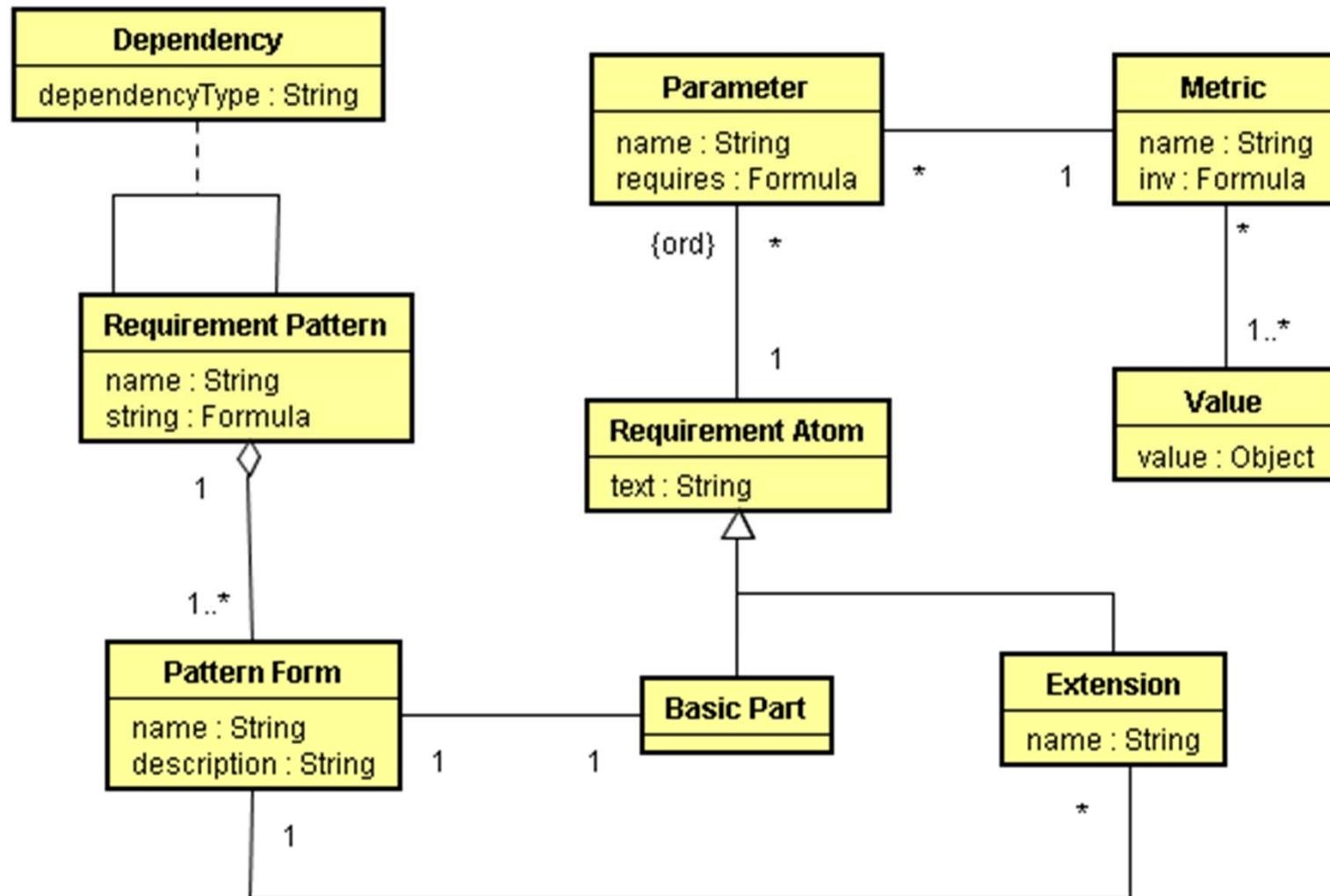
## Applicability cond.:

Basic Form + (Local OH[0..1] + Full OH[0..1])[0..1] +  
OH Technology[0..1]

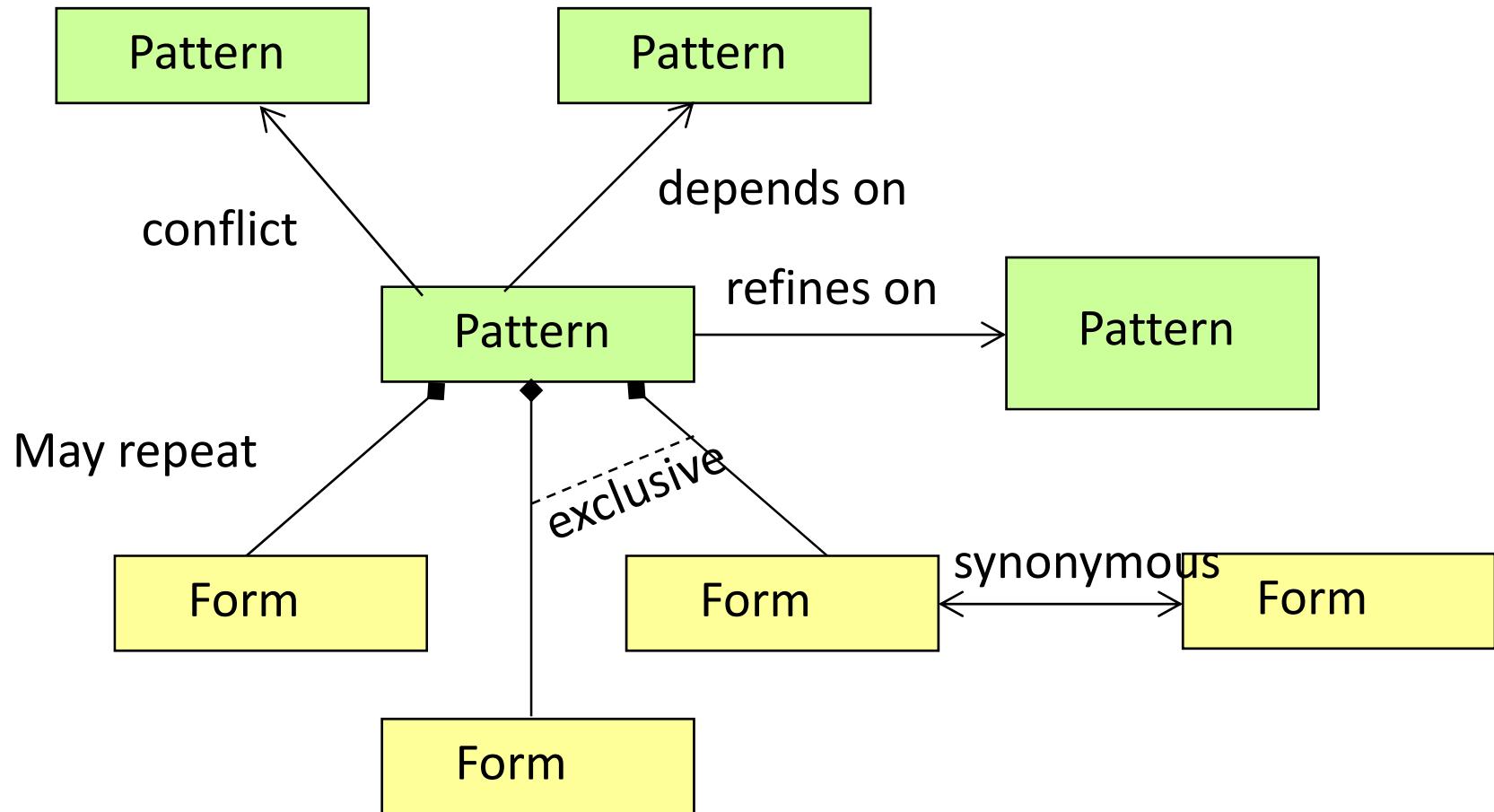
# Pattern dependencies

- Express how patterns are interrelated
- Usual types of dependencies
  - Depends
  - Conflicts
  - Synergies
  - Refines

# Final structure of an SRP



# Structure of the catalogue



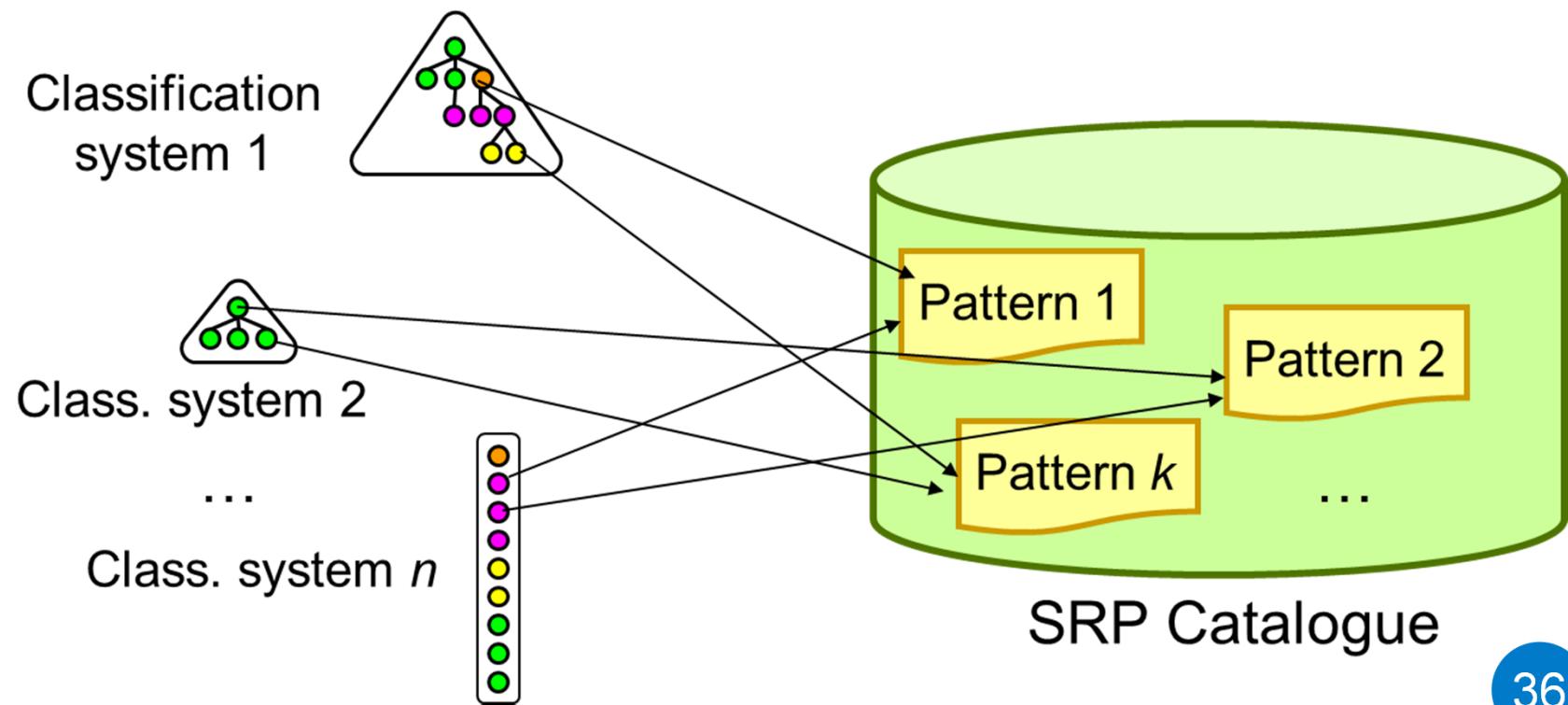
# Classification schemas



Group of Software Engineering for Information Systems  
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# SRP catalogues

- Stores the set of patterns
- Key issue: how is it organized → classification schema
  - We look for adaptability → not fixed



# Quality models

- A quality model is:

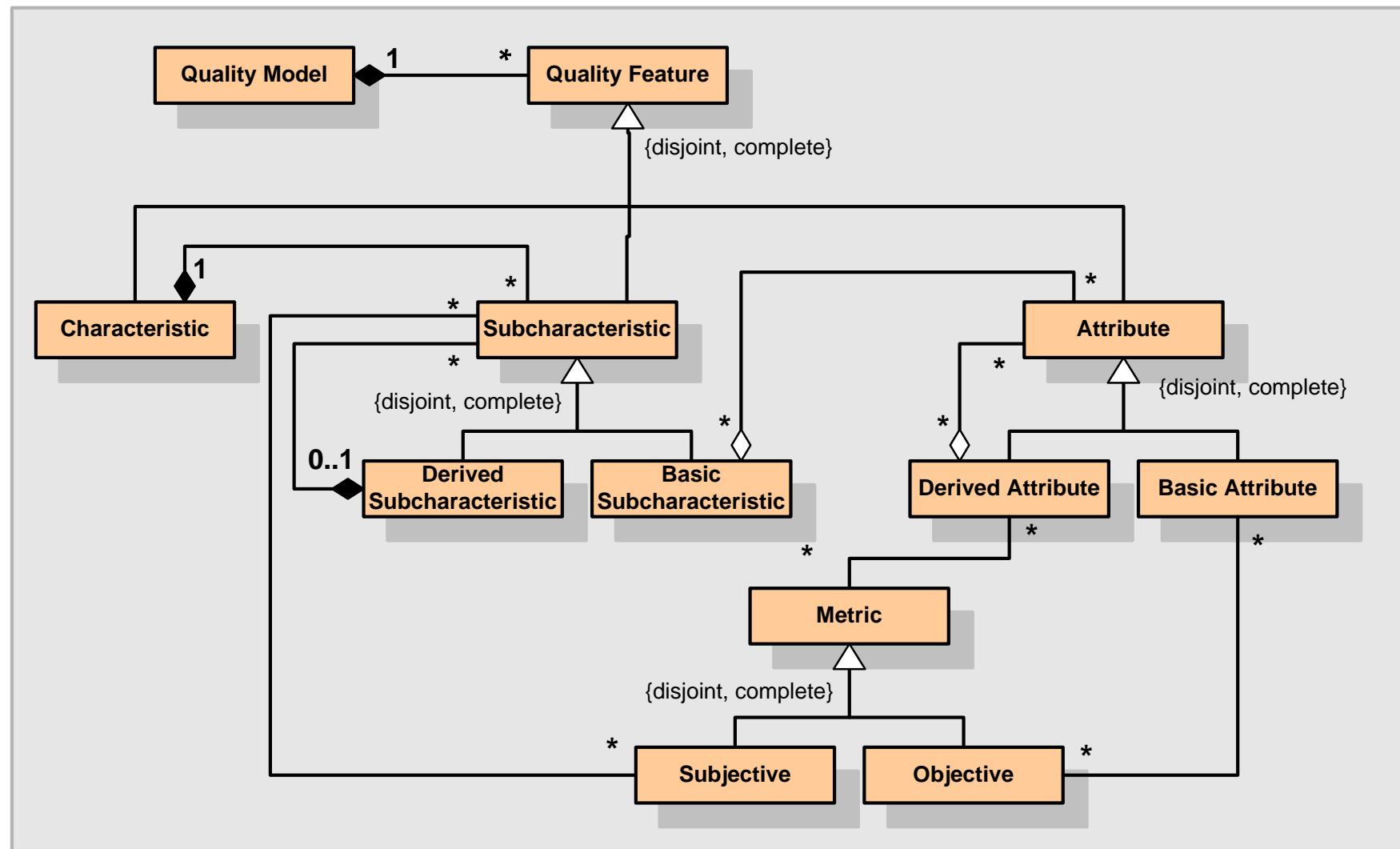
“The set of characteristics and the relationships between them which provide the basis for specifying quality requirements and evaluating quality”

ISO/IEC 14598-1, Software product evaluation –  
Part 1: General overview. 1999.
- These characteristics may be used as classifiers for the catalogue

# ISO/IEC 25010 standard

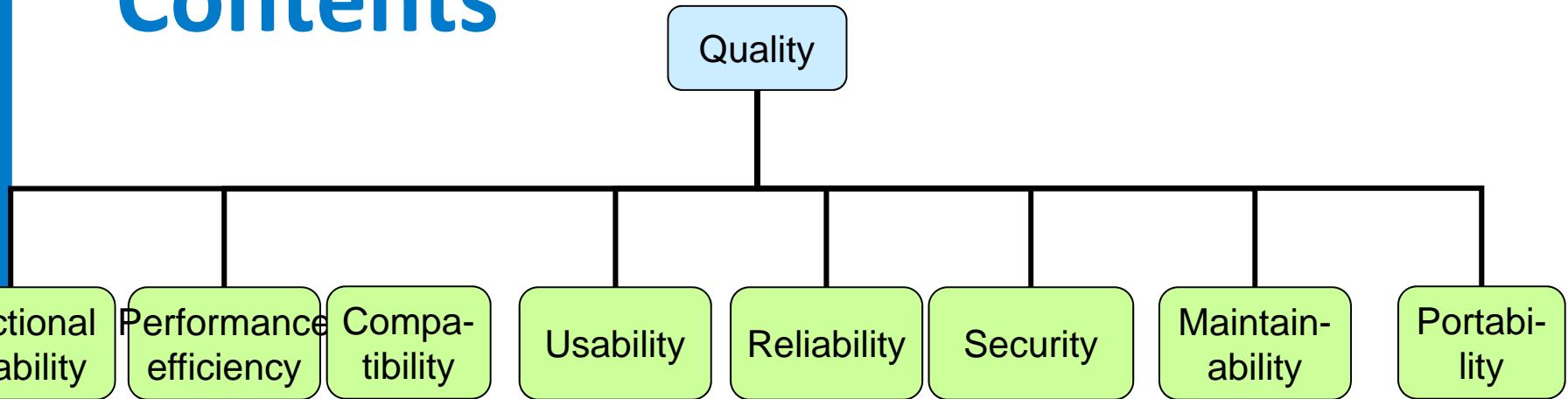
- Evolution of the former ISO/IEC 9126-1 (2001)
- Characteristics
  - Hierarchical
  - Extensible
  - Widespread

# General structure



Botella et al. “ISO/IEC 9126 in practice: what do we need to know?”. SMEF’04.

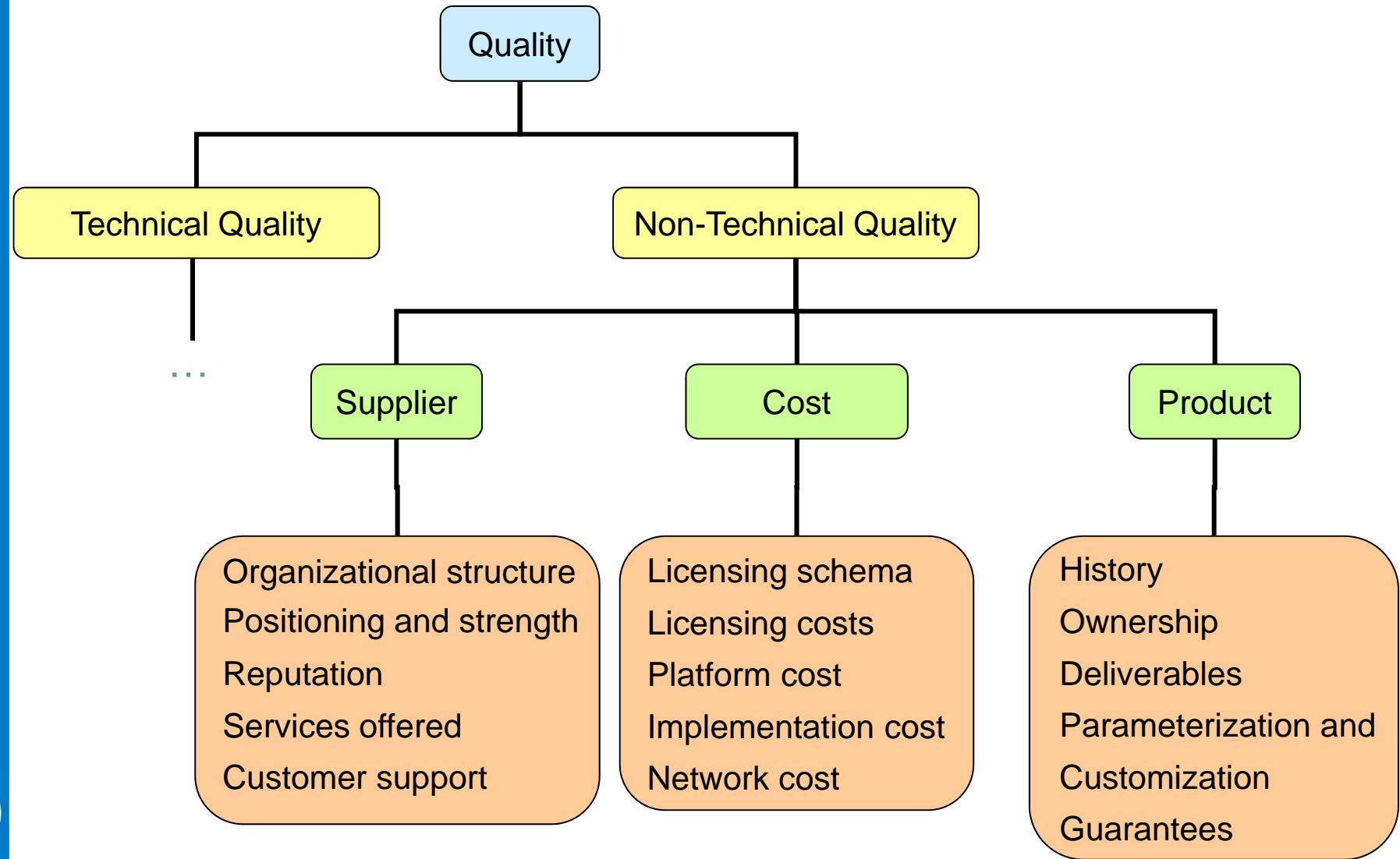
# Contents



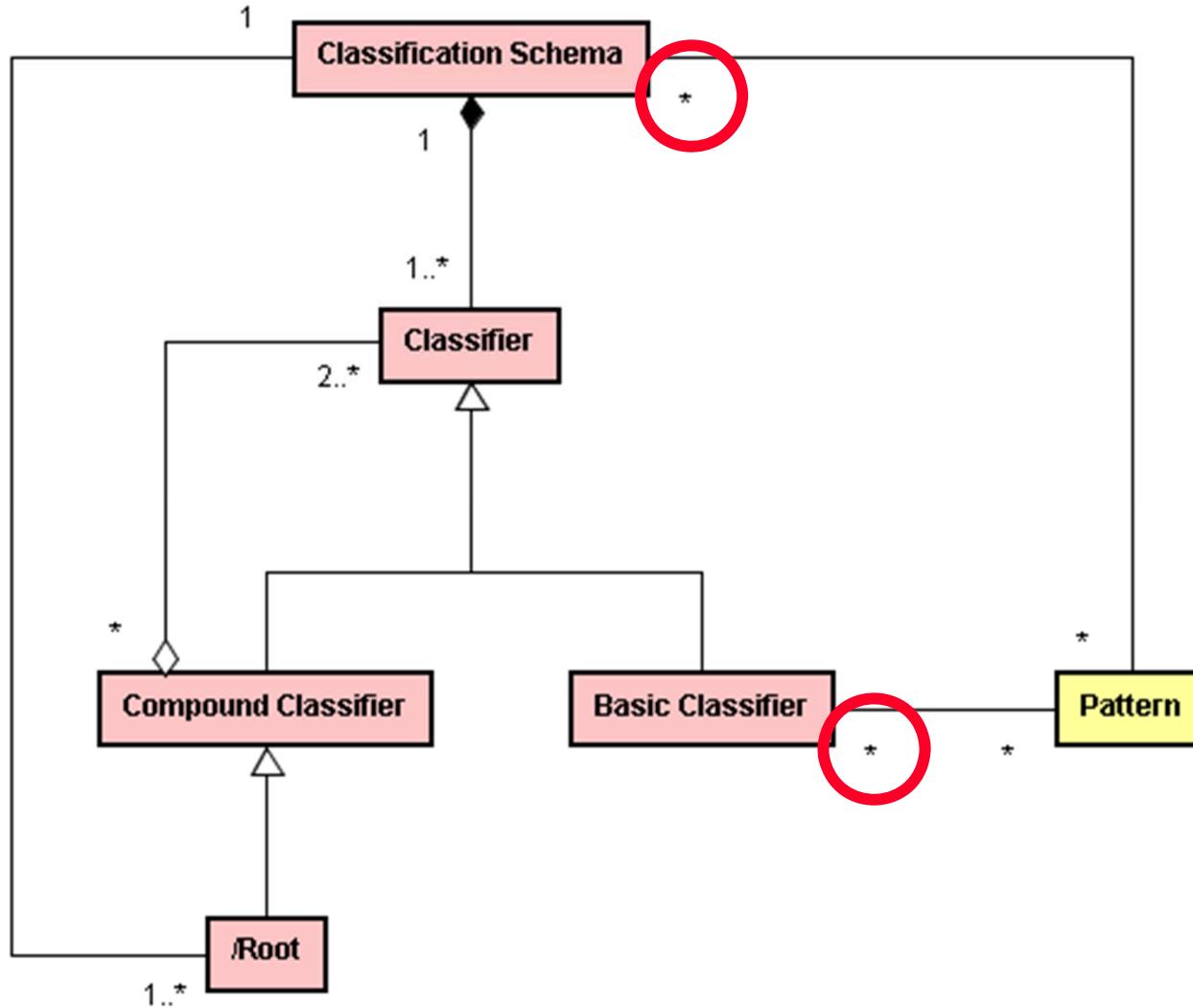
# Non-technical factors

- Non-technical factors are not in the scope of the standard
- Option 1: distribute them
  - Interfere in the categorization
  - It may be difficult to place some of them
- Option 2: add them
  - Use the principles behind the standard
  - 3 new characteristics and 15 subcharacteristics

# Enlarged catalogue



# SRP catalogues



# Construction process



Group of Software Engineering for Information Systems  
UNIVERSITAT POLITÈCNICA DE CATALUNYA

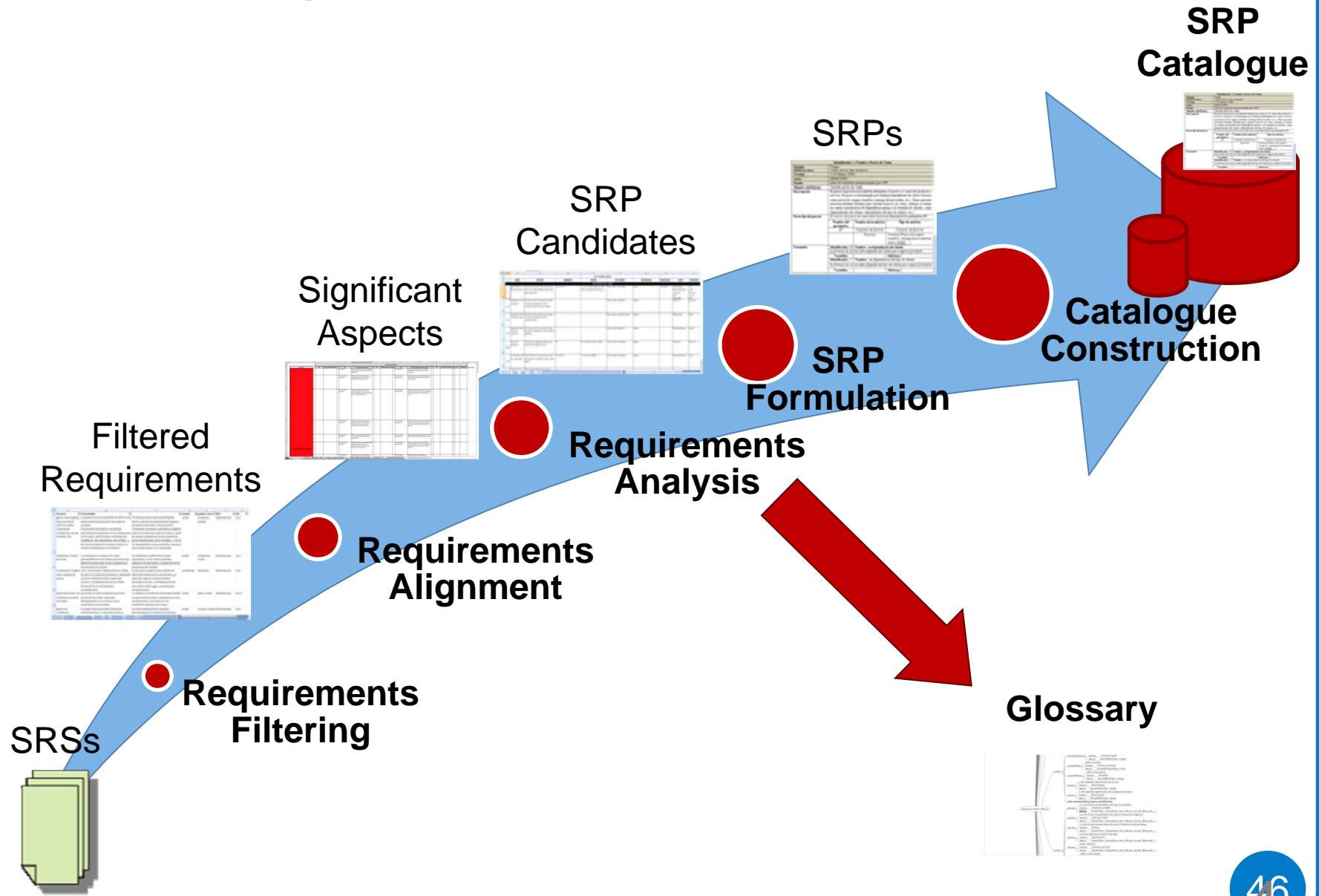
# Each catalogue is a different world

Each context is different

In general:

- Functional requirements: highly domain-dependent
  - General-purpose SPR to be very general
  - Each domain may refine them
- Non-functional and non-technical requirements
  - The percentage of reuse grows

# Catalogue Construction Process



# Alignment

Simple is beauty

A	B	C	D	E	F	G	
1	Resumen	Funcionalidad	modulo	palabras clave	libro	Ref	
3	gestion de prospectos, datos de clientes, perfil de compra, historicos de instalaciones, tipo de contratos, etc	Le système fournira la possibilité de définir et de sélectionner dynamiquement des cibles de prospects. Il fournira les informations nominatives, quantitatives et subjectives sur les contacts et/ou sur les clients, (profil d'achat, historiques des installations, des interventions, des contrats,...), afin de leur proposer de nouveaux produits ou services correspondant à leurs besoins.	El sistema proporcionará la posibilidad de definir y seleccionar dinámicamente objetivos de clientes potenciales. Proporcionará la información nominativa, cuantitativa y subjetiva sobre los contactos y/o sobre los clientes, (perfil de compra, antecedentes de las instalaciones, de las intervenciones, de los contratos...), con el fin de proponerles nuevos productos o servicios que corresponderán a sus necesidades.	ventas	prospectos, clientes	Electrosecurity	4.2.2
4	estadisticas y analisis de ventas	Les statistiques et analyses des ventes paramétrables et multi critères apporteront des éléments de description et de compréhension des évolutions du marché.	Las estadísticas y análisis de las ventas esqueléticas y multi criterios aportarán elementos de descripción y comprensión de las evoluciones del mercado.	ventas	estadisticas, ventas	Electrosecurity	4.2.4
5	la facturacion integrara varios metodos de calculo	Enfin, la facturation intégrera plusieurs modes de calcul du montant des échéances, notamment suivant le nombre de forfaits optionnels souscrits. Les dépassements de ces forfaits donneront lieu à une facturation complémentaire.	la facturación integrará varios métodos de cálculo del importe de los vencimientos, en particular, según el número de delitos opcionales suscritos. Los rebasamientos de estos delitos darán lugar a una facturación complementaria.	contabilidad	facturacion	Electrosecurity	4.2.8
6	precio de articulos por modificacion al precio de compra	Les articles en stock commercial auront leur propre tarif de ventes, réactualisé périodiquement ou en fonction d'une modification du prix d'achat.	Los artículos en existencias comerciales tendrán su propia tarifa de ventas, actualizada de nuevo periódicamente o en función de una modificación del precio de compra.	ventas	precio, ventas	Electrosecurity	4.2.12
	gestion de vendedores,	Le système devra permettre d'alimenter automatiquement un calcul des primes et	El sistema deberá permitir abastecer automáticamente un cálculo de las primas y	ventas	comision, ventas	Electrosecurity	4.2.4

# Some obvious alignment

The system must be available 22 hours per day and 7 days per week. The system should not stop more than 1 hour per working day. The solution's availability rate should be 98% minimum.

The system must be available 10 hours per day and 5 days per week. The system should not stop more than 10 minutes per working day. The solution's availability rate should be 98% minimum.

# Some not-so-obvious alignment

The solution should permit the trace of the modifications and access to the encrypted data in the database. The essential data to trace are: user name, date, accessed or modified data.

The solution should permit to trace all the user actions. The data to trace are: user name, date, accessed or modified data.

All the operations made over a type of object (dossier, document files, etc.) may be traced

The solution should permit to trace all the access to protected data. The data to trace are: user name, date, accessed or modified data. An historical record of this accesses must be maintained

# Searching candidates

The problem of deciding if a requirement is (part of) an instance of an SRP is the key in the catalogue construction:

- Some requirements may be too specific
- Others may be a particularization of some general case

The effort of identifying, writing, finding, applying and maintaining an SRP must be exceeded by the benefit of their use



# Guide to creating an SRP (1)

1. Decide if it is worth to create
  - How often is it expected to be used?
  - What value does it deliver when applying it?
  - How much work will give to write it and maintain it?
2. Create the skeleton of the pattern
  - Name, goal, classification
3. Write some examples of application
  - Based on existing SRS

# Guide to creating an SRP (2)

## 4. Analyse in detail the examples

- They are “imperfect” instances
- What are their commonalities?
- What may be improved?
- Remove everything that does not belong to the SRP

## 5. Identify the parameters

- Flexibility vs. Complexity
- Write information about the values
  - If numerical, typical ranges or constraints
  - If enumerated, typical labels

# Guide to creating an SRP (3)

## 6. Write the templates

- Identify possible alternatives and variants
  - Synonymous, ...
  - Extensions, particularizations, ...
- Identify possible properties
  - Multiplicity, ...
- Perfection is not effective
  - The catalogue will be naturally adjusted over time

# Guide to creating an SRP (4)

## 7. Validate

- Use departing SRS
- Make some piloting
- Peer review

## 8. Closing the pattern

- Write all missing information

## 9. Catalogue update

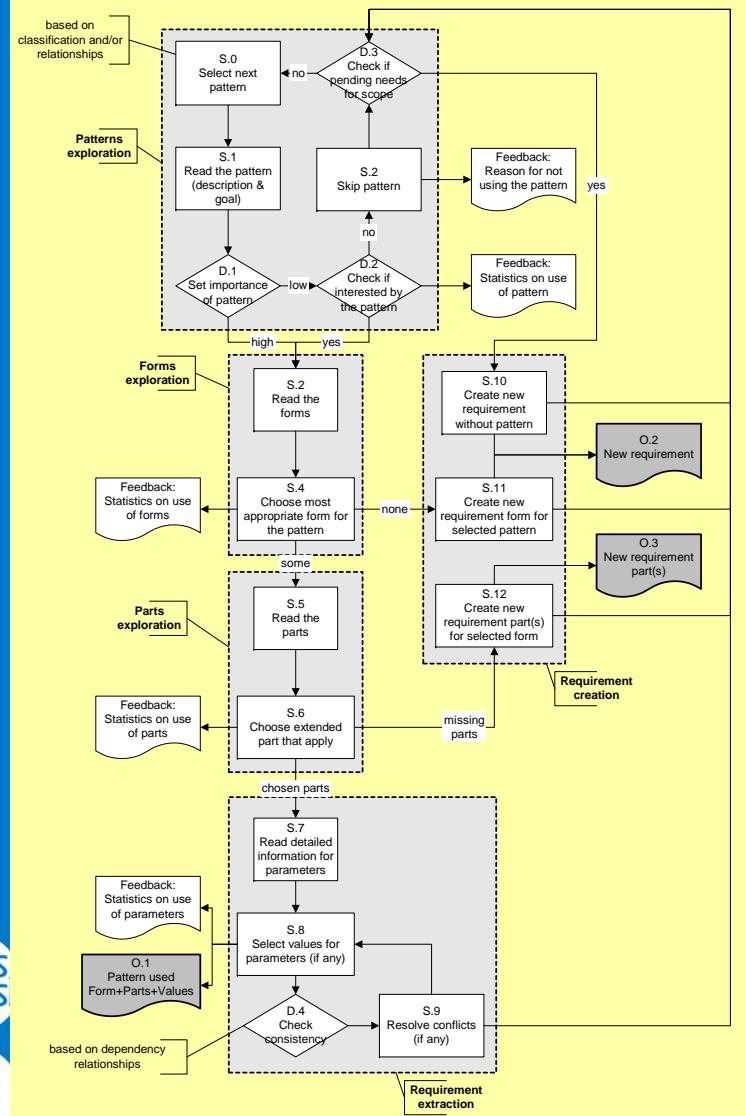
- Store in the catalogue
- Update the network of relationships (conflicts, synergies, ...)

# Application process



Group of Software Engineering for Information Systems  
UNIVERSITAT POLITÈCNICA DE CATALUNYA

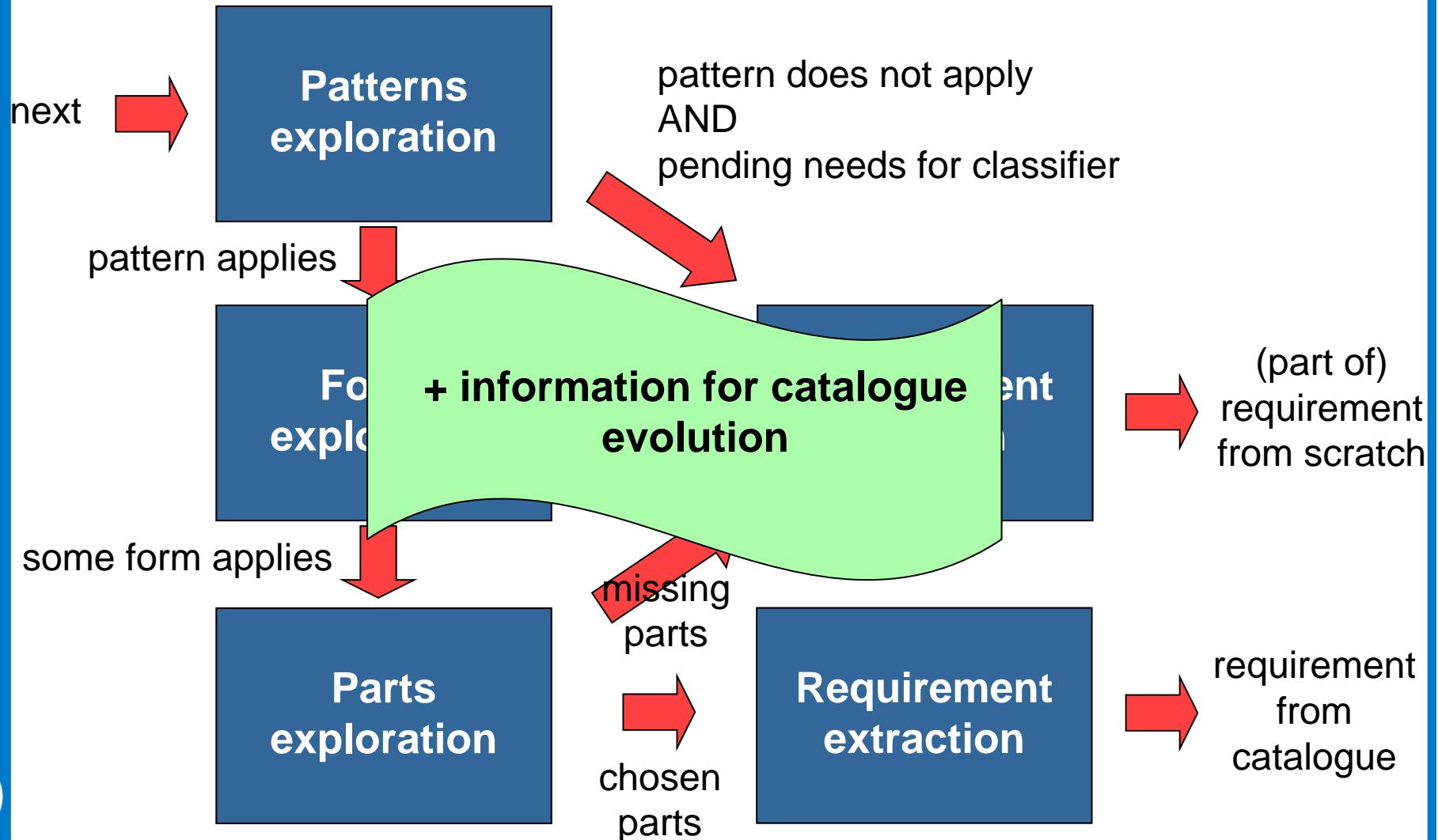
# Application process



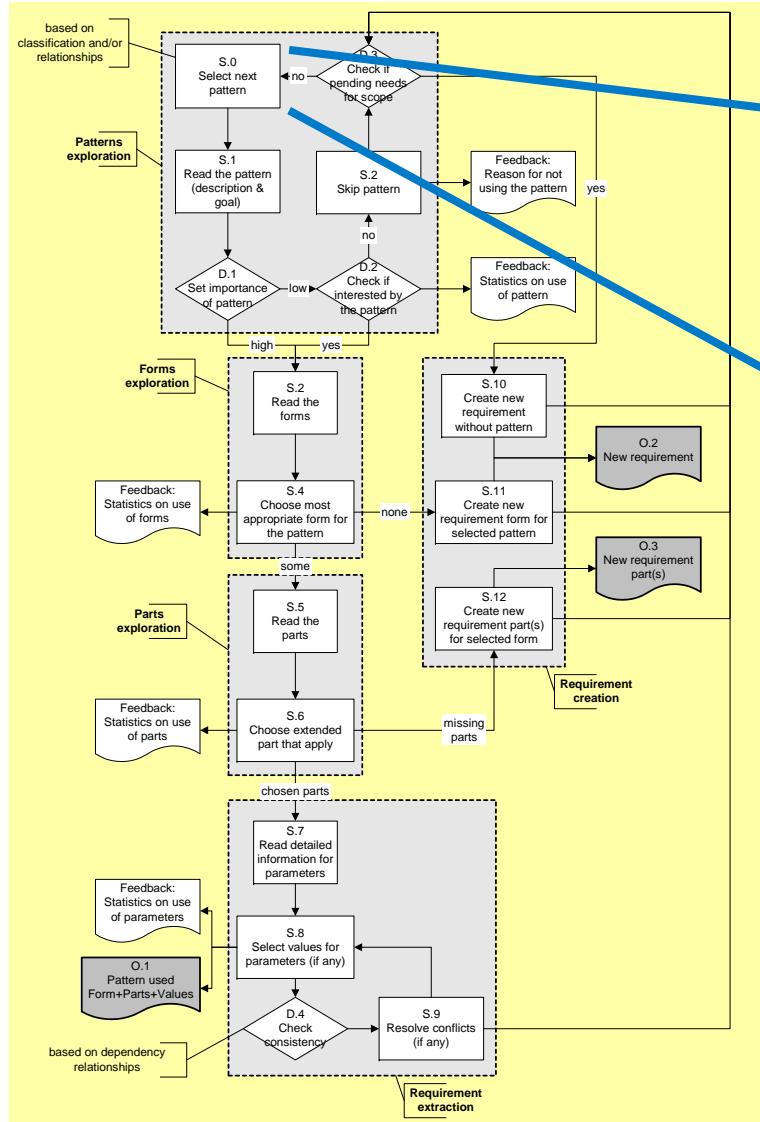
## ▶ Assumptions:

- May be patterns missing
- Not-automatic process
- Atomicity of information
- Complete classification schemas
- Process guided by one classification schema
- Catalogue existence made transparent to the customer

# High-level view



# Navigation



# Select next pattern

- classification schema
  - dependencies among patterns
  - keywords
  - others: qualifiers...

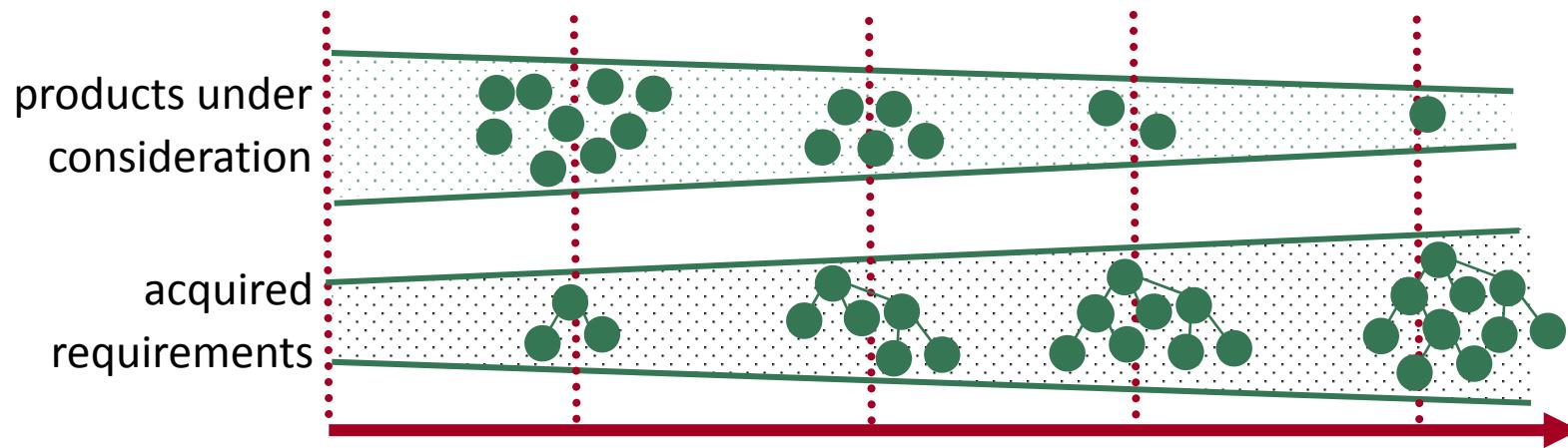
# Analysis of an existing catalogue: the CRPHT catalogue



Group of Software Engineering for Information Systems  
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# OTS selection processes

Classical OTS Selection Processes overlap RE and Component Screening



Iteration nicely captures the idea of reconciling requirements and actual market offering

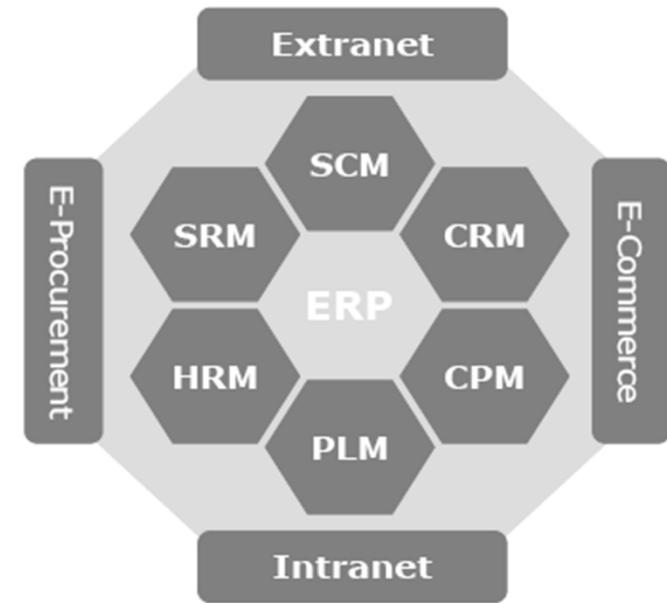
But it does not fit well in call-for-tender processes

# Call-for-tender processes

OTS selection processes conducted by a public document that contains the requirements and evaluation rules of the system-to-be

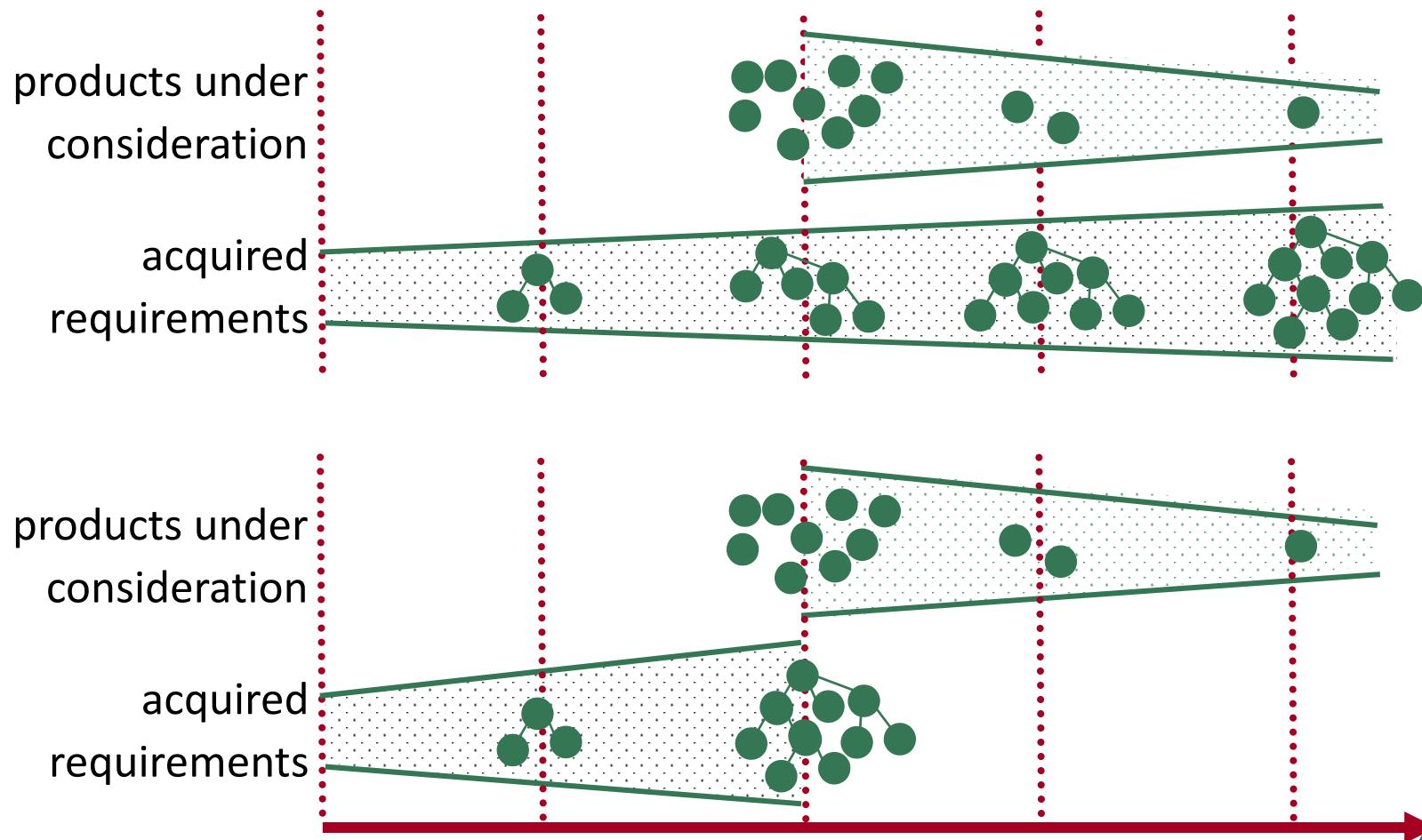
Usually:

- ▶ Public administrations
- ▶ Great impact on the organization
- ▶ Transparency is a must
- ▶ Involve coarse-grained OTS components (ERP, CRM, SCM, ...)



# Call-for-tender processes

Call-for-tender processes are different:



# Collaboration

SSI department, Centre de Recherche Publique Henri Tudor (CRPHT)

## Work

- Helping SME with no experience in Requirements Engineering.
- Designing SRS to conduct Call-For-Tender processes for selecting OTS solutions.

## Sample of SSI's requirements

- |   |   |
|---|---|
| 1. The system must be available 22 hours per day and 7 days per week.   | 2. Should be possible to use the system in English or French.   |
| 3. The system should not stop more than 1 hour per working day. The solution's availability rate should be 98% minimum. | 4. The solution should permit to trace all the user actions. The data to trace are: user name, date, accessed or modified data. |

## State

- More than 40 projects done.
- Applied reuse: starting a new project by editing the most similar one.

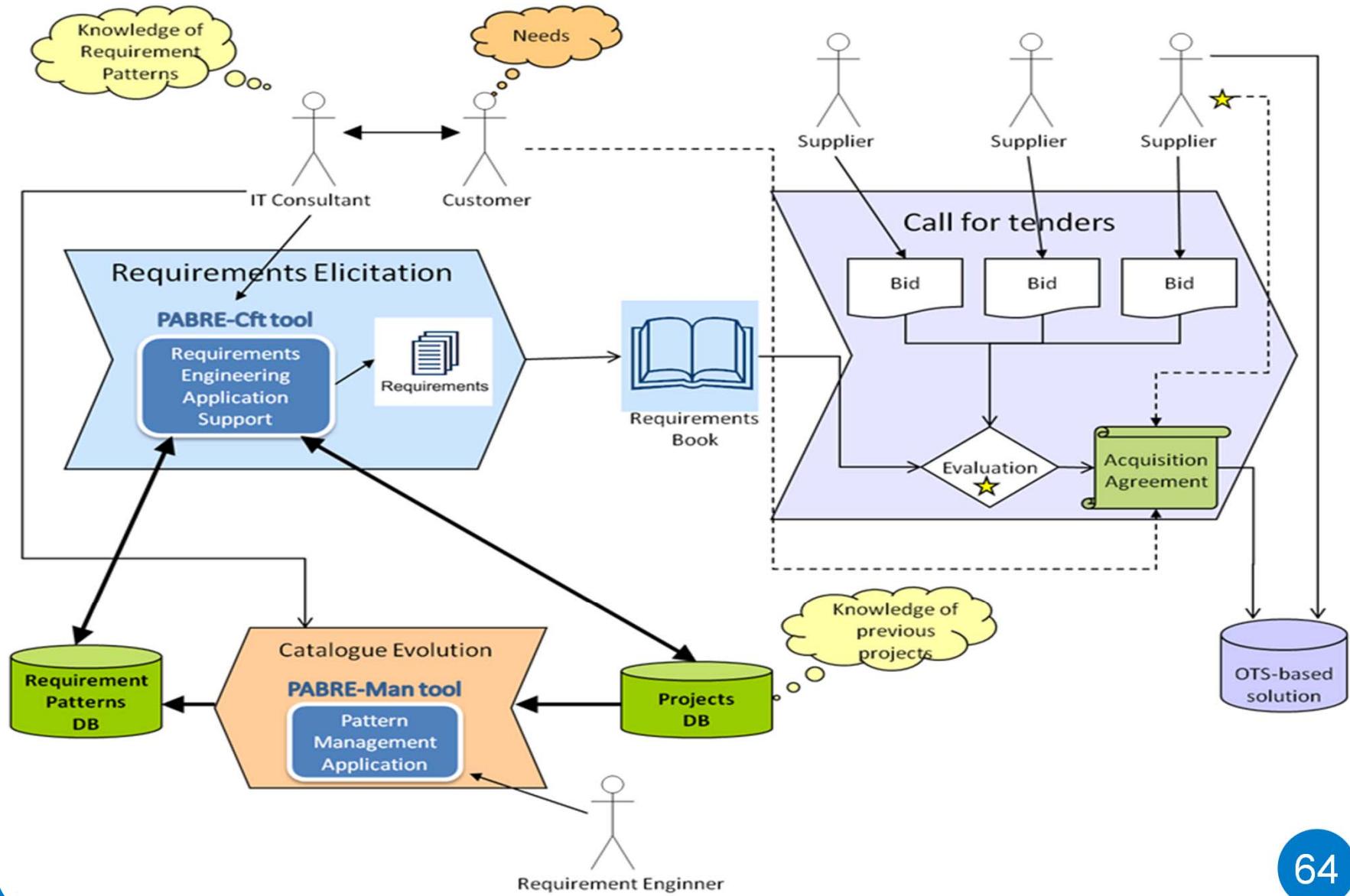
## Need

- Better capitalization of requirements in a high-level manner (like Patterns).
- Requirement books contains ambiguous, incomplete or incoherent requirements.

## Strategy

- Software Requirements Patterns (SRP)

# Overall picture



# Example of non-technical SRP

**SRP: Supplier Economic Information**

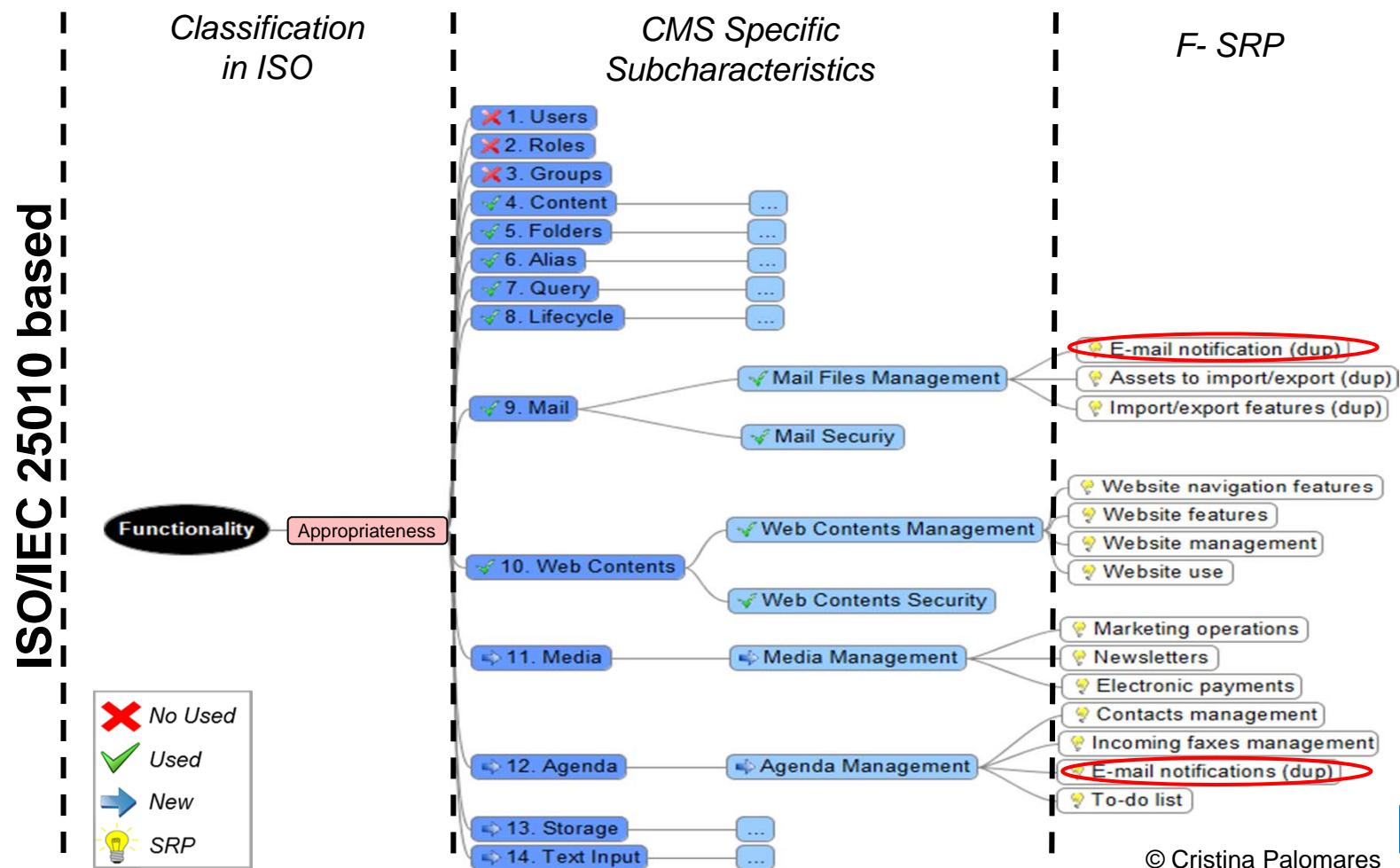
**Goal: Assessing the economic situation of the supplier**

*Does the Customer requires specific conditions on the economic situation of the supplier?*

SRP Form							
 <b>No</b>	 <b>Yes</b>						
SRP Form							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="background-color: #4f81bd; color: white; text-align: center;">Economic Situation Information Form</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"> <b>SRP Parts</b> <ul style="list-style-type: none"> <li>• <b>Fixed:</b> The supplier shall provide economic information of its company.</li> <li>• <b>Extended 1:</b> The supplier shall provide information of its company's turnover on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> <li>• <b>Extended 2:</b> The supplier shall provide information of its company's net income on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> </ul> </td> <td style="vertical-align: top;"> <b>SRP Parts</b> <ul style="list-style-type: none"> <li>• <b>Fixed:</b> The supplier shall fulfill some economic situation prerequisites.</li> <li>• <b>Extended 1:</b> The supplier's company shall have a minimum net income of <u>amount</u> <u>currencyUnit</u> on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> <li>• <b>Extended 2:</b> The supplier's company shall have a minimum turnover of <u>amount</u> <u>currencyUnit</u> on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> </ul> </td> </tr> <tr> <td style="vertical-align: top;"> <b>Constraints</b> <ul style="list-style-type: none"> <li>• <b>Fixed</b> part cannot be applied more than once.</li> <li>• <b>Extended 1</b> and <b>Extended 2</b> parts cannot be applied more than once.</li> </ul> </td> <td style="vertical-align: top;"> <b>Constraints</b> <ul style="list-style-type: none"> <li>• <b>Fixed</b> part cannot be applied more than once.</li> <li>• <b>Extended 1</b> and <b>Extended 2</b> can be applied more than once if they are applied with disjoint values for the parameters (<u>amount</u>, <u>amountOfTime</u>).</li> <li>• <b>Soft Constraint.</b> <b>Extended 1</b> and <b>Extended 2</b> applications should use the same <u>currencyUnit</u> and <u>timeUnit</u>.</li> </ul> </td> </tr> </tbody> </table>		Economic Situation Information Form		<b>SRP Parts</b> <ul style="list-style-type: none"> <li>• <b>Fixed:</b> The supplier shall provide economic information of its company.</li> <li>• <b>Extended 1:</b> The supplier shall provide information of its company's turnover on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> <li>• <b>Extended 2:</b> The supplier shall provide information of its company's net income on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> </ul>	<b>SRP Parts</b> <ul style="list-style-type: none"> <li>• <b>Fixed:</b> The supplier shall fulfill some economic situation prerequisites.</li> <li>• <b>Extended 1:</b> The supplier's company shall have a minimum net income of <u>amount</u> <u>currencyUnit</u> on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> <li>• <b>Extended 2:</b> The supplier's company shall have a minimum turnover of <u>amount</u> <u>currencyUnit</u> on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> </ul>	<b>Constraints</b> <ul style="list-style-type: none"> <li>• <b>Fixed</b> part cannot be applied more than once.</li> <li>• <b>Extended 1</b> and <b>Extended 2</b> parts cannot be applied more than once.</li> </ul>	<b>Constraints</b> <ul style="list-style-type: none"> <li>• <b>Fixed</b> part cannot be applied more than once.</li> <li>• <b>Extended 1</b> and <b>Extended 2</b> can be applied more than once if they are applied with disjoint values for the parameters (<u>amount</u>, <u>amountOfTime</u>).</li> <li>• <b>Soft Constraint.</b> <b>Extended 1</b> and <b>Extended 2</b> applications should use the same <u>currencyUnit</u> and <u>timeUnit</u>.</li> </ul>
Economic Situation Information Form							
<b>SRP Parts</b> <ul style="list-style-type: none"> <li>• <b>Fixed:</b> The supplier shall provide economic information of its company.</li> <li>• <b>Extended 1:</b> The supplier shall provide information of its company's turnover on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> <li>• <b>Extended 2:</b> The supplier shall provide information of its company's net income on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> </ul>	<b>SRP Parts</b> <ul style="list-style-type: none"> <li>• <b>Fixed:</b> The supplier shall fulfill some economic situation prerequisites.</li> <li>• <b>Extended 1:</b> The supplier's company shall have a minimum net income of <u>amount</u> <u>currencyUnit</u> on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> <li>• <b>Extended 2:</b> The supplier's company shall have a minimum turnover of <u>amount</u> <u>currencyUnit</u> on the last <u>amountOfTime</u> <u>timeUnit</u>.</li> </ul>						
<b>Constraints</b> <ul style="list-style-type: none"> <li>• <b>Fixed</b> part cannot be applied more than once.</li> <li>• <b>Extended 1</b> and <b>Extended 2</b> parts cannot be applied more than once.</li> </ul>	<b>Constraints</b> <ul style="list-style-type: none"> <li>• <b>Fixed</b> part cannot be applied more than once.</li> <li>• <b>Extended 1</b> and <b>Extended 2</b> can be applied more than once if they are applied with disjoint values for the parameters (<u>amount</u>, <u>amountOfTime</u>).</li> <li>• <b>Soft Constraint.</b> <b>Extended 1</b> and <b>Extended 2</b> applications should use the same <u>currencyUnit</u> and <u>timeUnit</u>.</li> </ul>						
Parameter Metrics							
<u>amount</u> , <u>amountOfTime</u> : integer ( <u>inv</u> integer must be greater than 0) <u>timeUnit</u> : domain {years, months, ...} <u>currencyUnit</u> : domain {USD, EUR, GBP, JPY, ...}							

# Functional SRP Catalogue for CMS

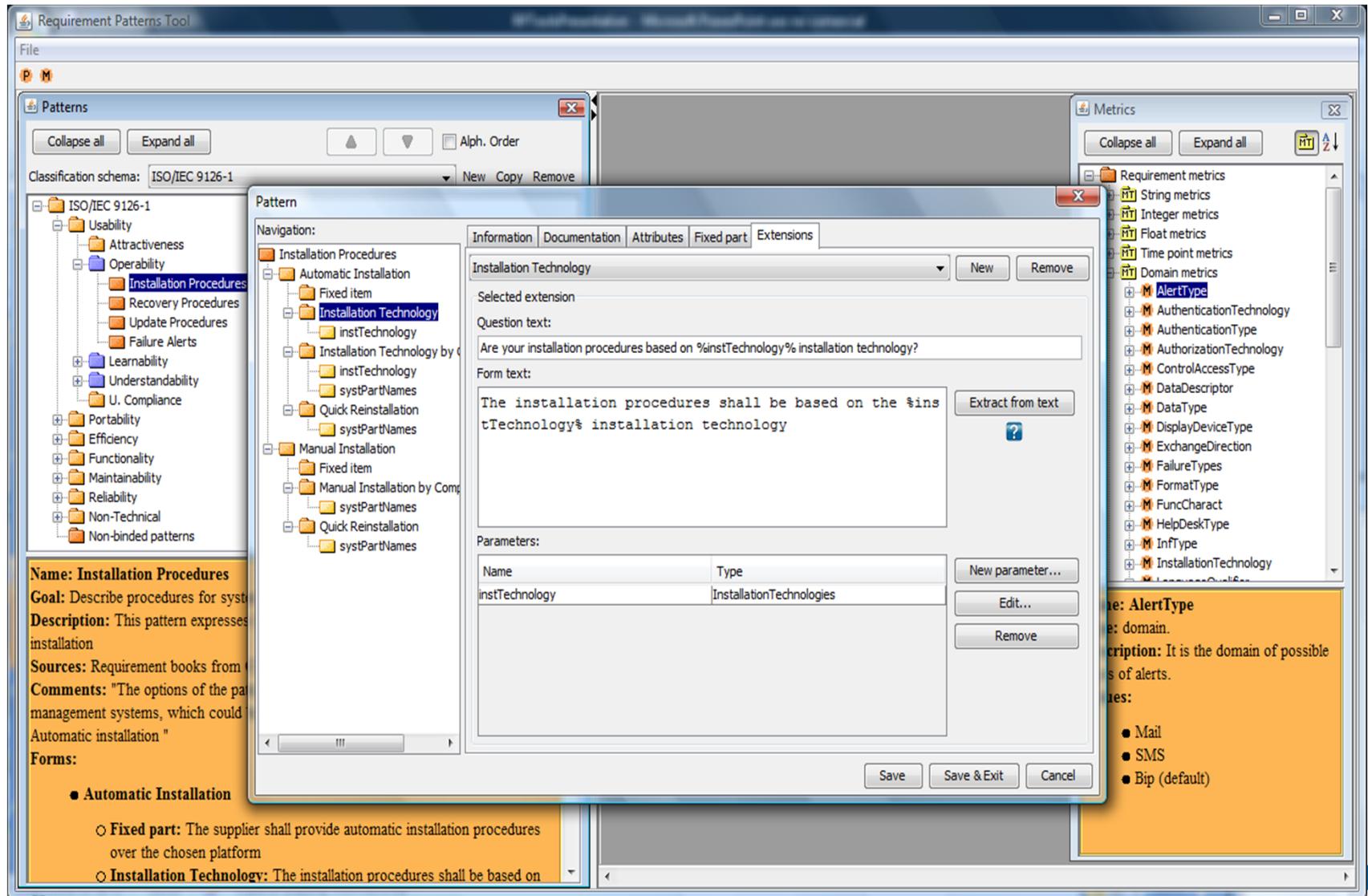
- 45 F-SRP obtained
- Classified using a CMS Classification Schema



# Example of functional SRP

Content Version Management (Solution Excerpt)	
Goal: Stating the functionalities of version management	
<i>Does the client need version management over all contents stored in the system?</i>	
	
<b>General Version Management Form</b> <ul style="list-style-type: none"> <li>• <b>Fixed:</b> The system shall manage versions of the stored contents.</li> <li>• <b>Automatic:</b> The system shall propose automatically the creation of new versions depending on the change done in the content.</li> </ul>	<b>Specific Version Management Form</b> <ul style="list-style-type: none"> <li>• <b>Fixed:</b> The system shall manage versions over specific stored contents.</li> <li>• <b>Versioned:</b> The system shall do versioning over <b>contentTypes</b> contents.</li> <li>• <b>Automatic:</b> The system shall propose automatically the creation of new versions over <b>contentTypes</b> depending on the changes done in the content.</li> </ul>
<b>Req. Templates</b>	<b>Req. Templates</b>
<b>Constraints</b>	<b>Constraints</b> <p>Fixed part (1) Extended parts: <i>Automatic Versions (0..1)</i></p> <p>Fixed part (1) Extended parts: <i>Versioned Contents (0..1)</i> <i>Automatic Versions (0..1)</i></p> <p>Consistency Rules:</p> <ul style="list-style-type: none"> <li>- <i>Versioned Contents must be applied before Automatic Version</i></li> <li>- <i>contentTypes(Automatic Versions) subset of contentTypes(VersionedContents)</i></li> </ul>
<input checked="" type="checkbox"/> <b>contentTypes</b> : set of contentType <ul style="list-style-type: none"> <li>◦ <i>contentType:</i> publications   reports   base documents...</li> </ul>	

# Tool support



# Conclusions



Group of Software Engineering for Information Systems  
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# SRP for the R-engineer toolkit

SRP provide structured abstractions:

- To guide the elicitation process
- To produce better quality SRS
- To facilitate easier validation
- To boost productivity of the RE process

Opens additional possibilities:

- Linking to architectural decisions
- Testing guidelines

# The End



Group of Software Engineering for Information Systems  
UNIVERSITAT POLITÈCNICA DE CATALUNYA