

A Catalogue of Non-Technical Requirement Patterns

Cristina Palomares, Carme Quer, Xavier Franch

*Universitat Politècnica de Catalunya
Barcelona, Catalunya, Spain
{cpalomares, cquer, franch}@essi.upc.edu*

Cindy Guerlain, Samuel Renault

*Public Research Centre Henri Tudor
Luxembourg, Luxembourg
{samuel.renault, cindy.guerlain}@tudor.lu*

Abstract—Software Requirement Patterns (SRP) have been proposed as an artifact for fostering requirements reuse. PABRE is a framework that promotes the use of SRP as a means for requirements elicitation, validation and documentation in the context of IT procurement projects. In this paper, we present a catalogue of non-technical SRP included in the framework and present in detail some of them. We also introduce the motivation to arrive to these patterns.

Keywords—software requirement patterns; requirements reuse

I. MOTIVATION

The work presented in this paper stems from the needs of the Public Research Centre Henri Tudor (TUDOR) at Luxembourg when conducting IT procurement projects over time. Since 2004, TUDOR works in collaboration with freelance and independent consultants. These consultants are federated in a business network that we refer as CASSIS. They are trained to innovative methods produced by research projects and they use these methods in industrial contexts. TUDOR monitors the application of these methods by consultants to ensure that they do not deviate over time.

One of the main methods delivered to consultants is a requirement engineering method used to design Software Requirements Specification documents (SRS) for IT procurement projects in small and medium size companies [1]. Consultants work in collaboration with customers to help them in identifying their needs for a new IT system supporting their business activities, and then selecting the most relevant system accordingly to their needs. In this particular context, requirements engineers' consultants define SRS for external customers and not for their internal purpose. Consultants' customers are usually looking both for an IT system and for its implementation. In other words, they have requirements towards an IT system and towards additional services. For this reason, the scope of the SRS often encompasses functional (F), non-functional (NF) [2] and non-technical (NT) [3] requirements. According to the empirical feedback of Tudor experimentation with public and private organisations, when selecting a business software solution, usually an organisation needs to address the three kinds of requirements. We have observed that F requirements are the most important in number and criticality, whilst NF and NT requirements are the most redundant or similar in between projects.

So far, consultants and TUDOR have performed more than 40 projects in compliance with that method. The initial approach for capitalising requirements knowledge among the consultants was quite basic. It consisted in reusing fragments of a former SRS as a basis to build the new SRS. This approach was simple to use but required to be aware of the former projects, which was not easy for the consultants due to their decentralized organisation in a business network.

The second TUDOR approach to capitalise requirements knowledge was to design SRS' templates based on existing SRS with similarities. This approach no longer requires the consultants to be aware of all former projects. However, the SRS' templates remained unstructured as domain experts built them both on their own knowledge and on assumptions of similarities found in existing SRS but without any underlying metamodel.

The limitations of these reuse approaches led TUDOR to collaborate with the Software Engineering for Information Systems research group (GESSI) at the UPC to define new artefacts, methods and techniques for requirements reuse. Specifically the PABRE framework [4] that arose from this collaboration is based in the use of Software Requirement Patterns (SRP) as those presented in this paper.

In Section II we show the main parts of the structure of the PABRE SRP by means of an example. Next, in Section III we present the overall structure of an NT SRP catalogue. The main part of the paper is Section IV where we introduce two examples of SRP following the workshop pattern template to structure them. Finally, some conclusions are given in Section V. It is not the aim of this paper to explain neither the SRP metamodel, nor the process for obtaining or method for using the catalogue in requirements elicitation processes, we refer to [4][5][6] for details in these aspects.

II. PABRE SRP STRUCTURE

We present the structure of PABRE patterns through an example, the *Economic Situation* pattern (see Fig. 1), that illustrates the structure of patterns and helps to understand the metamodel behind them [5].

An SRP is a pattern that, when applied, produces software requirements related to the objective (goal) of that pattern. Applying the *Economic Situation* SRP we may produce requirements related to the goal of *Assessing the economic situation of the supplier* that procures a software system, as could be the supplier company's turnover or net incomes.

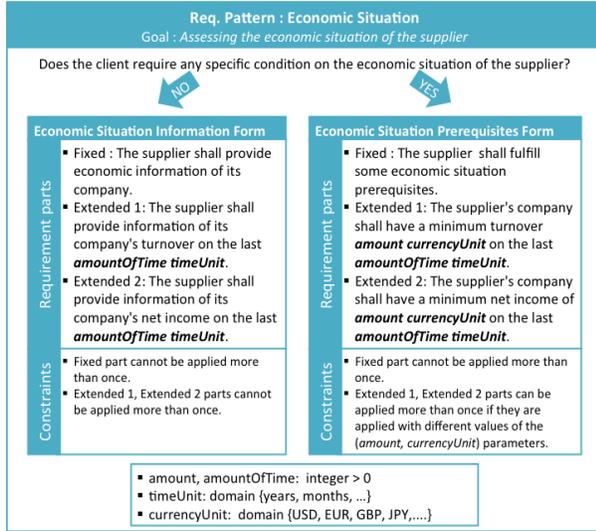


Figure 1. Economic Situation Pattern

A goal can be achieved in different ways. An SRP consists of several *Forms*, each one representing a different *solution* for achieving the goal. In the *Economic Situation* SRP, its goal can be attained by asking the supplier the relevant economic information (*Economic Situation Information* form), or by setting conditions or prerequisites on the economic situation that the supplier should have (*Economic Situation Prerequisites* form).

We organize Forms into Parts, each of them being a template. Each Form is characterized by a Fixed Part which states the minimal requirement that always holds when applying that form, and some Extended Parts which may be applied or not. The Fixed Part always becomes a requirement when an SRP is applied with this Form. Extended Parts are only used if more precise information is required in the specification. Due to this nature, the Fixed Part is usually quite generic and hardly measurable. For instance, the first form of Economic Situation is *The supplier shall provide economic information of its company*, whilst the two extended parts identify the type of information required (company's turnover or net income) and the period of time.

Usually, fixed and extended parts must conform to some Part Constraint represented by means of a regular expression that may involve some predefined operators (e.g., for declaring multiplicities or dependencies among parts, as Excludes and Requires). In the *Economic Situation* SRP, each part of the forms may be used just once in a specification project, and neither excludes nor requires dependencies among them.

From a syntactic point of view, both fixed and extended parts are similar. They are composed by the text to be used as a requirement and optionally some parameters to be instantiated when applying the pattern. Parameters establish their *Metric*, eventually a correctness condition *inv*, and also may be *related* to other parameters (belonging to other patterns) such that they must have the same value. The

second form in the *Economic Situation* SRP declares two extended parts that identify additional conditions on this form. For example, the second extended part allows stating prerequisites on the net supplier incomes (by assigning values to the parameters *amount* and *currencyUnit*, e.g. 1M EUR) for a certain period of time (by assigning values to the parameters *amountOfTime* and *timeUnit*, e.g. 2 years). The metrics and the correctness conditions of the parameters are detailed at the bottom of the figure.

III. THE NT SRP CATALOGUE

The NT SRP were obtained after mining 6 SRS from past projects conducted by TUDOR and the consultants. They are a part of a bigger catalogue, which currently contains other 29 NF patterns and 47 F patterns that apply on the Document Management Systems domain.

The SRP are organized and classified in the catalogue by means of one or more schemas, and give different views to the consultants for facilitating its browsing during requirements elicitation. The idea is to provide to different people with different background a different view of the catalogue with which they are used (see Fig. 2). For instance, TUDOR, and its trained consultants, have their own requirements classification, whilst the GESSI team usually works with the ISO/IEC 9126-1 standard [7].

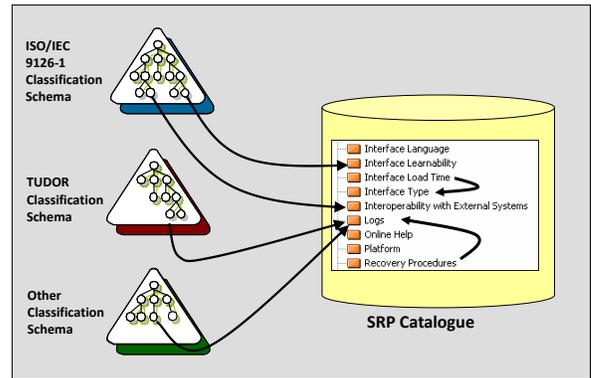


Figure 2. SRP Catalogue Classification Schemas

In this paper we organize the catalogue using an ISO 9126-based schema. However, since in this standard the characteristics and subcharacteristics do not address non-technical aspects of software, we used the NT-ISO/IEC 9126 catalogue that we proposed in previous works [3]. This extension adds 3 characteristics (*Supplier, Business* and *Product*) and 15 subcharacteristics to the standard. Before classifying the NT SRP according to this schema, some changes had to be done to take into account some differences on the use of the catalogue.

On the one hand, initially that catalogue was created to include the criteria to assess the quality of a final software product, whereas the NT SRP state requisites for the procurement of a system (probably by gluing or adapting several products). This is the reason why we needed to add a new characteristic to group the SRP about the

implementation project: the *Project* characteristic, decomposed into two subcharacteristics: *Business Scheduling* and *Supplier Relationships*.

On the other hand, some related subcharacteristics were merged into just one. Specifically, they were those related to the cost of the business. The original subcharacteristics were too static: *Licensing Costs*, *Platform Costs*, *Implement Costs* and *Network Costs*, but the new subcharacteristic integrates all these costs in a cost breakdown structure allowing the flexibility to add new ones.

Table I shows the resulting classification. It is worth to mention that most of the subcharacteristics have some NT SRP bound which is an indicator that the projects used as baseline data were comprehensive enough.

TABLE I. NT SRP CLASSIFICATION ACCORDING TO NT-ISO/IEC 9126

1. Supplier		NT SRP	
1.1 Organizational Structure		• Supplier Administrative Information	
		• Supplier Organization	
		• Supplier History	
1.2 Positioning and Strength		• Supplier Economic Information	
		• Supplier Workforce	
1.3 Reputation		• Supplier Business Experience	
		• Supplier Quality Certification	
1.4 Services Offered		• Training	
1.5 Support		• Maintenance Procedure	
		• Type of Maintenance	
2. Product			
2.1 History		• Product History	
		• Community Support	
2.2 Deliverables		• Delivered Documentation	
		• Source Code	
2.3 Parameterization and Customization		-----	
3. Business			
3.1 Licensing Schema		• -----	
3.2 Ownership		• Intellectual Property Rights	
3.3 Guarantees		• Warranty	
3.4 Costs		• Cost Breakdown Structure	
4. Project			
3.8 Business Scheduling		• System Implementation Scheduling	
		• Progress Control	
		• Project Management Method	
		• Final acceptance	
		• Release	
		• Analysis	
		• Data Migration	
		• Development	
3.9 Supplier Relationships		• Acceptance Tests	
		• Steering Committee	
		• Meetings Organization	
		• Access to Customer Premises	
		• Privacy	
		• Progress Control	
		• Quality Assessment	
		• Payment Procedure	
		• Settlement of Disputes	
		• Supplier People Assigned to the Project	
		• Help Desk	
• Crash Response			

IV. THE NT SRP CATALOGUE

We present here two of the NT SRP of our catalogue: *Intellectual Property Rights* (Tables II and III) and *Quality Assessment* (Tables IV and V). Tables II and IV show the SRP general attributes using the RePa workshop template. Tables III and V describe the SRP detailed solution according to the model presented in Section II.

The Intellectual Property Rights SRP (Table II) is suitable in procurement projects (*RE Activity*) where the customer wants to state the property right over of different deliverables obtained as a result of the system implementation project (*Problem*, which in our framework is named goal). The *Forces* in the customer organization influence in the *Application* of the SRP *Solution*, specifically in choosing which *Parts* and *Parameter values* to use.

TABLE II. "INTELLECTUAL PROPERTY RIGHTS" PATTERN SUMMARY

Pattern Template	
Name	Intellectual Property Rights.
Authors	GESSI, TUDOR
Context	RE Activity. Elicitation, Specification
	Pattern Type. Product
	Business Domains. Domain independent
	Organizational Environment Factors • Customer organization wants to establish the property rights of assets generated.
Stakeholders	Customer, Supplier, Customer Legal Department, Supplier Legal Department
Problem	Need of setting the property rights over the different deliverables obtained as a result of the system implementation project
Forces	<ul style="list-style-type: none"> • The customer can be interested in the property of some deliverables. • The customer can be interested in have some rights of use on the deliverables.
Solution	The solution is to apply the pattern as detailed in Table III, adding the corresponding requirements in a SRS document.
Application	The whole process is described in [2] <ul style="list-style-type: none"> • Browse the pattern, • Check if the problem/goal is relevant for the context, • Choose the most appropriate form, • Extract the fixed part, • Check and extract the most relevant extended parts taking into account the dependencies • Choose the parameter values taking into account the dependencies, • Add requirements in the specification.
Known Uses	• IT procurement projects
Cataloguing	Classification NT-ISO/IEC 9126: Business: Ownership
	Related Patterns <ul style="list-style-type: none"> • Quality Assessment (dependency with the values of parameter: projectDeliverables) • Delivered Documents (dependency on the values of parameter: documentType)

TABLE III. NT SRP "INTELLECTUAL PROPERTY RIGHTS": DETAILED SOLUTION

Intellectual Property Rights			
Goal: Stating the rights of using deliverables result of the system implementation project			
Description: This pattern expresses the need of setting the property rights over the different deliverables result of the system implementation project			
Keywords: Project Deliverable, Project Asset, Intellectual Property, Property Rights			
Requirement Form <i>Intellectual Property Rights</i>	Description	This form expresses the need of setting the property rights over the different deliverables obtained as a result of the system implementation project	
	Constraints	Fixed part application: (1..*) Extended parts application: <i>Project Deliverable Use (*)</i> ; <i>Project Assets Return (0..1)</i> Parameter values constraints: actor, projectDeliverables DisjointValues (Fixed Part, Project Deliverable Use)	
	Fixed Part	Form Text	At the end of the system implementation project, projectDeliverables shall become property of the actor .
		Param	Metric
		<i>projectDeliverables</i> is a non-empty set of the different assets related to a system implementation project (e.g., hardware, software, documents, data, etc.)	ProjectDeliverables = Set(ProjectDeliverable) ProjectDeliverable = Domain (hardware, software, data and documents paid by customer as project deliverables, etc.)
		<i>actor</i> represents one of the possible roles related to a system implementation project (usually the customer or the supplier)	Actor = Domain (supplier, customer, etc.)
	Extended Part <i>Project Deliverable Use</i>	Form Text	At the end of the system implementation project the actor can use IPRConditions the projectDeliverables
		Param	Metric
		<i>actor</i> as above	<i>Actor</i> as above
		<i>projectDeliverables</i> as above	<i>ProjectDeliverables</i> as above
		<i>IPRConditions</i> represents the intellectual property rights over some project deliverable for some actor	IPRConditions = Set (IPRCondition) IPRCondition = Domain (freely, with no restriction, with non-commercial use, with respect of License contract, etc.)
	Extended Part <i>Project Assets Return</i>	Form Text	The supplier shall return the projectAssets that the customer provided him just for the system implementation.
		Param	Metric
		<i>projectAssets</i> is a non-empty set of the different assets related to a system implementation project (e.g. customer business process documents, customer company business reports, etc.)	ProjectAssets = Set(ProjectAsset) ProjectAsset = String (e.g. customer business process documents, customer company business reports, etc.)

In the detailed solution (Table III), the constraints of use declare that, when applying the pattern, the fixed part can be used several times in a project (1..*), provided that the values of the attributes *actor* and *projectDeliverables* are different in each use. Thus, different deliverables may be property of different actors.

The first extended part helps to state the rights on *projectDeliverables* that are not property of an actor. This part can also be used more than once (*), where uses are constrained by the same rule, i.e. it is not possible to have the same combination of parameters' values in different uses of the fixed and first extended part. The semantic behind the rule is that it is not possible to be the owner of a deliverable and require rights on that deliverable.

The second extended part allows stating the return of the assets that the supplier borrowed from the customer after finishing the project.

An example of application of this SRP in an IT project:

- At the end of the system implementation project, **the software developed, the delivered documents (except the installation manuals) and the drivers** shall become property of the **customer**.
- At the end of the system implementation project, **the hardware, the installation manuals and**

development software shall become property of the **supplier**.

- At the end of the system implementation project, the **customer** can use **paying some quota the hardware and the installation manuals**.
- At the end of the system implementation project, the **supplier** can use **freely** the delivered documents.
- The supplier shall return the **customer business process documents and the documentation of the substituted software system** that the customer provided him just for the system implementation.

The SRP related to Intellectual Property Rights are: *Quality Assessment* and *Delivered Documents* (see *Related Patterns* in Table II). Both relationships are dependencies on the values of the parameters of each pattern. The idea is that if a certain statement of ownership or rights of use is required on a project deliverable, maybe it is necessary to stand the level of quality that such deliverable shall have, and if the deliverable is a document it should be among the documents delivered in the project (*Delivered Documents*).

The Quality Assessment SRP (Table IV) is suitable in procurement projects where the customer wants to state its right of performing quality assessment of the supplier or of the project deliverables. The *Forces* influence in the

Application of the SRP Solution: depending on whether the customer wishes to do the assessment by some specific quality criteria or based on some quality standard a different *Form* will be applied. In the second case, the second form *Quality Standard-based Assessment* shall be used.

In the detailed solution (Table V), the constraints of use state that in the application of the pattern the fixed part has to be used once (1), and all the extended parts can be or not applied (0..1), except the *Deliverables Quality Assessment* extended part that can be used several *times* (*) for different sets of project deliverables (see the *Parameter values constraints*). One application of the SRP could be:

- If the customer considers it necessary, during the system implementation project, s/he shall be allowed to assess the quality of the process or project deliverables taking into account a quality standard.
- The quality of the *software design documentation* shall be assessed taking into account the **IEEE 1016** quality Standard.
- The quality of the *requirements specification document* shall be assessed taking into account the **IEEE 830** quality Standard.
- The customer shall establish the subset quality standard criteria to be applied **before December 2012**.

The SRP related to the Quality Assessment SRP are (Table IV): *Supplier Quality Certification*, *Intellectual Property Rights*, *Delivered Documents*. The first relationship is a dependency with the SRP *Supplier Quality Certification*: if a customer is interested in the *Quality Assessment* SRP s/he will be also interested in the *Supplier Quality Certification* SRP. The other two are dependencies on the values of the parameters *projectDeliverables* and *documentType* of the SRP *Intellectual Property Rights* and *Delivered Documents* respectively.

V. CONCLUSIONS

In this paper we have presented the overall structure of a set of non-technical SRP integrated in the PABRE framework, that are part of the catalogue constituted by 113 SRP (functional, non-functional and non-technical). We have illustrated the details of NT SRP by showing in detail two particular patterns. Besides the catalogue, the PABRE framework is built upon a pattern application process [4], a metamodel [5] and tool support [8]. Currently, the catalogue is available under a specific Creative Commons license that allows its use in non-commercial way (e.g. for research and experimentation purpose) but without derivative works [6].

The value of the approach has been qualitatively identified as positive from first practitioners' feedback collected in a survey addressed to practitioners who produced the SRS documents on which we based to build the approach. We are currently setting-up the experiment with practitioners that will bring enough data for quantitative analysis of the value of the approach.

Our future work will consist on progressing with the validation and evolution of the NT SRP catalogue and the supporting tools. Also, we aim at adopting techniques to help in the detection of SRP in SRS documents.

TABLE IV. "QUALITY ASSESSMENT" PATTERN SUMMARY

<i>Pattern Template</i>	
<i>Name</i>	Quality Assessment.
<i>Authors</i>	GESSI, TUDOR
<i>Context</i>	RE Activity. Elicitation, Specification
	Pattern Type. Product
	Business Domains. Domain independent
	Organizational Environment Factors Customer organization that gives importance to the quality assessment practices.
<i>Stakeholders</i>	Customer, Supplier, Customer Quality Department, Supplier Quality Department, Supplier Project Management Office
<i>Problem</i>	Need of setting the customer right for performing quality assessment of the supplier or the project deliverables.
<i>Forces</i>	The customer can be interested or not in a certain standard for assessing the quality of software.
<i>Solution</i>	The solution is to apply the pattern, as in Table V, adding the corresponding requirements in an SRS.
<i>Application</i>	The whole process is described in [2] (see Table III for more details)
<i>Known Uses</i>	• IT procurement projects
<i>Cataloguing</i>	Classification NT-ISO/IEC 9126: Project: Supplier relationships
	Related Patterns
	<ul style="list-style-type: none"> • Supplier Quality Certification (the customer that applies Quality Assessment can be also interested in this pattern) • Intellectual Property Rights (dependency with the values of parameter: projectDeliverables) • Delivered Documents (dependency on the values of parameter: documentType)

ACKNOWLEDGMENT

This work has been partially supported by the Spanish project TIN-2010-19130-C02-01.

REFERENCES

- [1] S. Renault, B. Barafort, E. Dubois, M. Krystkowiak. Improving SME Trust into IT Consultancy: a Network of Certified Consultants Case Study. EuroSPI 2007.
- [2] K. Pohl. *Requirements Engineering: Fundamentals, Principles, and Technique*. Springer, 2010.
- [3] J. P. Carvallo, X. Franch, C. Quer: Managing Non-Technical Requirements in COTS Components Selection. RE 2006.
- [4] S. Renault, O. Mendez-Bonilla, X. Franch, C. Quer: A Pattern-based Method for building Requirements Documents in Call-for-tender Processes. IJCSA 6(5), 2009.
- [5] C. Palomares, C. Quer, S. Renault, F. De Lazzar. A Metamodel for Software Requirement Patterns. REFSQ 2010.
- [6] <http://www.upc.edu/gessi/PABRE/>
- [7] ISO/IEC 9126-1-2001 Software Engineering Product Quality Part 1: Quality Model, International Standards Organization, 2001.
- [8] C. Palomares, C. Quer, X. Franch. PABRE-Man: Management of a Requirement Patterns Catalogue. RE 2011.

TABLE V. NT SRP "QUALITY ASSESSMENT": DETAILED SOLUTION

QUALITY ASSESSMENT				
Goal: Stating the customer's right of performing quality assessment				
Description: This pattern expresses the need of setting the customer right for performing quality assessment of the supplier or of the project deliverables.				
Keywords: Quality, Quality assessment, Quality criteria, Quality standard				
Requirement Form <i>General Quality Assessment</i>	Description	This form expresses the need of setting the customer right for performing quality assessment of the supplier or the project deliverables regarding to specific customer quality criteria.		
	Constraints	Fixed part application: (1) Extended parts application: <i>Review Focus (0..1)</i> <i>Quality Criteria Agreement (0..1)</i>		
	Fixed Part	Form Text	If the customer considers it necessary, during the system implementation project, s/he shall be allowed to assess the quality of the process or the projectDeliverables .	
		Param	Metric	
		projectDeliverables is a non-empty set of the different products delivered during the system implementation project (e.g. hardware, software, documents, etc.)	ProjectDeliverables = Set(ProjectDeliverable) ProjectDeliverable = Domain (hardware, software, data and documents provided or paid by customer as project deliverables, etc.)	
	Extended Part <i>Review Focus</i>	Form Text	The customer shall focus the quality assessment on the qualityAspects .	
		Param	Metric	
	qualityAspects is a non-empty set of the different quality aspects to be assessed	QualityAspects = Set(QualityAspect) QualityAspect = Domain (specific development, treatment of the reported abnormalities, quality procedures, etc.)		
Extended Part <i>Quality Criteria Agreement</i>	Form Text	The customer shall agree with the supplier the level of quality expected for the various project deliverables.		
Requirement Form <i>Quality Standard-based Assessment</i>	Description	This form expresses the need of setting the customer right for performing quality assessment of the supplier or of the project assets regarding a quality standard.		
	Constraints	Fixed part application: (1) Extended parts application: <i>Process Quality Assessment (0..1)</i> <i>Deliverables Quality Assessment (*)</i> <i>Quality Criteria Establishment(0..1)</i> <i>Quality Criteria Agreement (0..1)</i> Parameter values constraints: <i>projectDeliverables, qualityStandard DisjointValues (Deliverables Quality Assessment)</i>		
	Fixed Part	Form Text	If the customer considers it necessary, during the system implementation project, s/he shall be allowed to assess the quality of the process or project deliverables taking into account a quality standard.	
	Extended Part <i>Process Quality Assessment</i>	Form Text	The quality of the process shall be assessed taking into account the qualityStandard quality Standard.	
		Param	Metric	
		qualityStandard represents the identifier of the quality standard that shall be used to assess the quality	QualityStandard = Domain (IEEE830, IEEE829, IEEE1016, ISO/IEC9126, ISO/IEC 15504-5, etc.)	
	Extended Part <i>Deliverables Quality Assessment</i>	Form Text	The quality of the projectDeliverables shall be assessed taking into account the qualityStandard quality Standard.	
		Param	Metric	
		projectDeliverables as above	ProjectDeliverables as above	
		qualityStandard as above	QualityStandard as above	
Extended Part <i>Quality Criteria Agreement</i>	Form Text	The customer shall agree with the supplier on the level of quality expected for the project deliverables.		
Extended Part <i>Quality Criteria Establishment</i>	Form Text	The customer shall establish the subset quality standard criteria to be applied timePreposition date .		
	Param	Metric		
	timePreposition represents the relationship with respect to a date	timePreposition = Domain (on, before, after, at, by...)		
	date is a time point representing the date in which the quality standard criteria shall be established	Date = TimePoint		