Using Business Rules in EXtreme Requirements

Maria Carmen Leonardi[1] and Julio Cesar Sampaio do Prado Leite[2]

[1] INTIA- UNCPBA - Tandil, Buenos Aires-Republica Argentina
`cleonard@exa.unicen.edu.ar`
[2] Pontifícia Universidade Católica do Rio de Janeiro — PUC-Rio
Rua Marquês de São Vicente 255, 22451-041 Gávea-RJ
Rio de Janeiro, Brasil
`julio@inf.puc-rio.br`

Abstract. Extreme Requirements (XR) is a proposal that tries to improve the quality of Extreme Programming (XP). XP is a well known agile method for software production. XP key elements are: little documentation, simplicity, analysis as constant activity, evolutionary design, integration and daily test. XR defines a requirements strategy that can be coupled with XP. In this article, we present an XR business rules based process. Our process is oriented to the customer, based on natural language, facilitating construction and validation. One of the strongest aspects of our proposed process is communication with customers, making them active participants in the software production process.

1 Introduction

Extreme Programming (XP) [1] is a development method that falls into the category of agile method, whose two main characteristics are to be adaptative and people oriented [2]. XP is based on oral communication, continuous testing and a code communication structure. XP uses descriptions, User-Stories, to register the desired system functionality. User-Stories are used in all XP cycles: requirements, design and test.

Research on software evolution [3] has convinced us that change is intrinsic to the task of software construction, so we need better processes to address change. We see Extreme Programming as a way of dealing with constant change. Because of this and due to its growing insertion on object-Oriented developments, we believe, from Requirements Engineering perspective, that it is important to adapt requirements strategies to enhance XP without altering its principles.

This paper is an extension of [4] where one of us proposes XR, a family of requirements processes based on scenarios. This article extends XR with the incorporation of a business rule model and a set of heuristics to derive CRCs cards [5].

In XP, customers[1] have an active role in the process, working in close contact with the engineers [2], providing business experience in which the software engineer will be able to trust and to consider it as the basis for the development. For this reason, we believe that it is fundamental to provide a customer-Oriented process to be used in the context of XP.

XR proposes the use of scenarios [6] and the Lexicon model (LEL) [7] to elicit and model requirements. LEL represents the language of the Universe of Discourse(*UofD*)[2], allowing a natural communication with clients. In this paper, we extend this approach to incorporate a business rule model [8]. Business rules are part of the heart of any organisation, modelling the policies and constraints that guide and govern the structure and behaviour of an organisation [9]. As XP deals with change, we think it is important to incorporate a Business rules based process, since it reflects how an organisation should behave and it is easily updated due to its granularity level. We also propose a set of heuristics to derive CRC cards[3] . Although more documents are added to the process, and one of XP goals is to try to reduce documentation, we consider it is useful to add an initial model for requirements with the objective of favouring communication with the customer. This documentation is obtained in a fast and easy way, with little overhead, without altering XP working philosophy and contributing to the clarity of the first stages of development by providing better communication with customers.

2 An Introduction to XP

Extreme Programming [1] is an agile method for small to medium software teams. It is a method designed for situations where the requirements are vague. Its key elements are: little documentation, simplicity, analysis as a constant activity, evolutionary design, integration and daily tests. It eliminates over documentation and focuses on small releases which are quickly implemented and tested. Therefore, the system grows together with the client's new knowledge and its new necessities. Figure 1 (http://www.extremeprogramming.org/map/project.html) shows the general process of XP.

In this work, we will focus on the first stages of XP. Nowhere in the XP process is there a mention to a requirements document, but the term requirements is used several times in the book [1]. XP uses User-Stories and CRCs[4] in the first stages of development, and we assume that these models are its requirements and design models.

[1] Despite the fact that XP refers to one customer (on-site customer), we understand that there is more than one customer, so we use the term customers (see Section 4.1).

[2] The Universe of Discourse is the overall context in which the software will be developed and operated [10].

[3] CRC (Class Responsibilities Collaborators) is an exploratory technique used in several object oriented methods during the design task. It is based on index cards that register the name of the class, its responsibilities and its collaborators. The CRC card was proposed by Beck and Cunningham [5], and used by the Responsibility Driven Design [11] and by the Fusion method [12] among others.

[4] Although CRC cards are not mentioned in the original XP proposal [1] they are referred to on the XP site [www.extremeprogramming.org] as product of the XP design process.

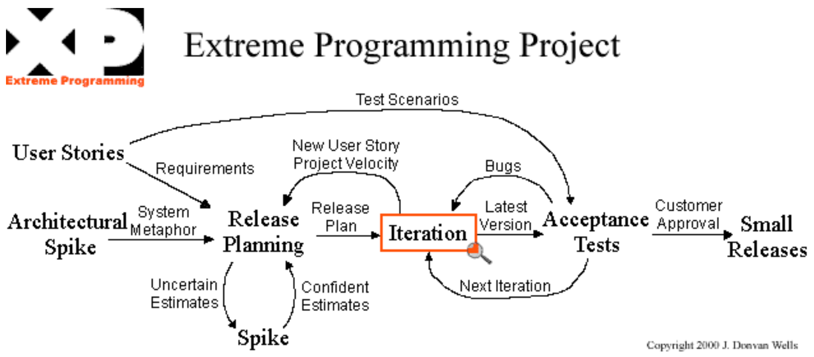


Fig. 1. XP Project (http://www.extremeprogramming.org/map/project.html)

The User-Stories or stories are written by the customer and report the tasks the system should perform. Its construction depends mainly on the ability of the client to define them. They are written in natural language, without a predetermined format, not exceeding some few text lines. Stories guide the construction towards the acceptance tests, key elements in XP, and they are used to estimate development time. In this sense, they only provide enough details to make a reasonable estimation. In the iteration planning, User-Stories are split up into tasks; the next step is a quick design session.

CRC cards [5] were proposed in order to document collaborative design decisions. The cards represent the responsibilities and collaborators of a particular class, focussing on the motivation for collaboration by representing many messages as a phrase of a text. XP uses CRC cards during design without imposing any particular discipline. Developers, managers, and users move the cards around making it easy to describe how classes work and interact. XP does not save the cards. The classes themselves are the documentation of the design: they represent what the system really is . In section 4.1 we describe some problems related to XP requirements and our proposal to deal with them. XR proposes the use of scenarios instead of User-Stories. Scenarios model improves the XP User-Stories techniques in several ways, but mainly due to its well established construction process [13]. We also supplement the XP requirements phase with client-Guided models and a simple construction process. Therefore, we improve communication with customers without altering the method principle of simplicity.

3 Natural Language Oriented Requirements Models

In this section, the models used in XR are briefly described. These models are exemplified in Section 5.

3.1 Language Extended Lexicon

The Language Extended Lexicon, LEL [7] is a structure that allows the representation of significant terms of the *UofD (Universe of Discourse)*. The purpose of the lexicon is to help the understanding of the application vocabulary and its semantics, leaving the comprehension of the problem for a future step. It unifies the language allowing communication with customers. . LEL is composed by a set of symbols which represents the basis of an application language. Symbols are, in general, words or phrases that customers repeat or emphasise. The lexicon is a series of symbols with the following structure: *symbol name:* word or phrase and set of synonyms, *notions* that describe the denotation of the symbol and *behavioural responses or impacts,* describing the symbol connotation. In the description of notions and impacts there are two basic rules that must be followed simultaneously: the "closure principle" that encourages the use of LEL symbols in other LEL symbols, thus forming a graph, and the "minimum vocabulary principle " where the use of symbols external to the application language is minimised, and the ones used should refer to a very small and well accepted general core. LEL terms define objects, subjects, verbal phrases and states.

The purpose of constructing a lexicon is not only to enable good communication and agreement between the customers and the engineering team but also to bootstrap the scenario and business rules construction process and to help in their description. The use of the symbols of the lexicon in the scenarios and business rules makes it possible for these symbols to be a natural hyperlink between these three representation structures, a fundamental characteristic of our Requirements Baseline concept [10]. The construction process is carried out by means of two elicitation techniques: interviews and reading. The first interviews are non-Structured and aim at collecting candidate symbols. Reading documents related to the application is also a way of finding symbols. After the first interviews and readings a list of symbols is defined, with the most frequently used words. Next, these symbols are classified according to a general classification. The following stage is the description of each symbol, consisting in determining its notions and behavioural responses, which then will be validated with customers. As a result of the validation interviews, we have the first LEL version, which will be polished to unify the syntax in order to attain a consistent and homogeneous LEL.

3.2 Scenario Model

A scenario describes *UofD* situations [6]. Scenarios use natural language as their basic representation. They are naturally connected to LEL. Figure 2 describes the components of a scenario.

Notions of *Constraint* and *Exception* are added to some of the components of a scenario. A *constraint* refers to non-Functional requirements and may be applied to context, resources or episodes. An *exception*, applied to episodes, causes serious disruptions in the scenario, asking for a different set of actions which may be described separately as an exception scenario.

The scenario construction process [13] starts from the application lexicon, producing a first version of the scenarios derived exclusively from the LEL. These scenarios are improved using other sources of information and organised in order to obtain a consistent set of scenarios that represents the application.. During or after these activities, the scenarios are verified and validated with the clients/users to detect Discrepancies, Errors and Omissions (DEO). The process is composed of five activities.

Title: identifies a scenario. In the case of a sub-scenario, the title is the same as the sentence containing the episode.
Objective: describes the purpose of the scenario.
Context: defines geographical and temporal locations and preconditions.
Resources: identify passive entities with which actors work.
Actors: detail entities actively involved in the scenario, generally a person or an organization.
Set of episodes: each episode represents an action performed by actors using resources. An episode may be explained as a scenario; this enables a scenario to be split into sub-scenarios.

Fig. 2. Scenario Components

1. The Derive activity aims at generating the derived candidate scenarios from the information of the LEL using the scenario model and applying the derivation heuristics. The derivation process consists in three steps: identifying the actors of the *UofD*, identifying candidate scenarios and creating them using the lexicon.
2. The Describe activity aims at improving the candidate scenarios by adding information from the *UofD* using the scenario model, the lexicon symbols in the descriptions and applying the description heuristics. The result is a set of fully described scenarios. This activity should be planned and usually relies on structured interviews, observations and document reading.
3. The Organise activity is the most complex and more systematised one in our scenario construction process. Its root is the idea of *integration scenarios*, "artificial" descriptions with the sole purpose of making the set of scenarios more understandable and manageable. Integration scenarios give a global vision of the application. Each integration scenario episode corresponds to a scenario.
4. The Verify activity is performed at least twice during the scenario building process, the first one over the fully described scenario set and the second after the Organise activity. This is done following a checklist with verification heuristics. As a consequence of this activity, two DEO lists are produced, one used at Describe and the other used during the LEL construction process. The verification is divided in intra scenarios, inter scenarios and against the LEL. Using the verification DEO lists, the scenarios and the LEL are modified. If major corrections were needed, a new verification could be required.
5. Finally, scenarios are validated with stakeholders usually by performing structured interviews or meetings.

3.3 Business Rule Model

We define business rules as statements about the enterprise's way of doing Business [8]. Organisations have policies in order to: satisfy the business objectives, satisfy customers, make good use of resources, and conform to laws or general business conventions. The business rule model distinguishes between functional rules and non-Functional rules as shown in Figure 3.

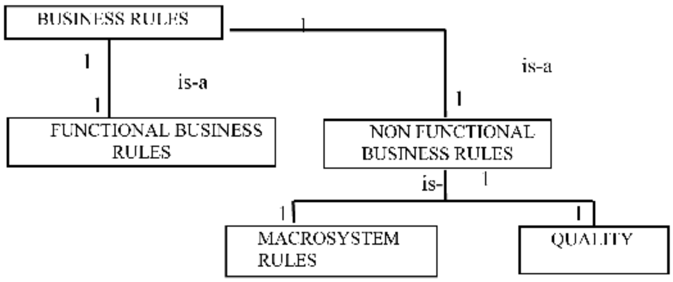


Fig. 3. Business rule Taxonomy

Functional rules are general policies regarding the organisation functionality. *Macrosystem rules* describe policies that constraint the behaviour and structure of the organisation. *Quality rules* are demands of the organisation on the characteristics of its processes or products. They usually reflect general policies related to quality standards or expectations of the organisation. We can follow some syntax patterns to build the rules, in which case their main components may be an LEL entry.

The business rules construction process [14] begins with the identification of the sources of information. Organisation documents, such as ISO required documents [15] and organisational models [16] [17] are generally the best sources. If the company does not have any documentation, other techniques such as observation, interviews and meetings should be used to acquire the information.

We categorise the sentences that appear in the sources considering their purpose in the organisation. We try to distinguish sentences referring to: limits, responsibilities and rights. To decide if a sentence is a business rule, we analyse if it is determined by a decision of the organisation (for internal or external reasons) or if it is an inherent sentence to the functionality of the *UofD* (in which case it is not considered a rule).

Taking the concept of stability [18] we determine the stability of each sentence. We use the degree of stability of a sentence not only to determine if it is a rule, but also to attach that information to the rule. Information regarding stability will help the construction of a change oriented architecture. Business rules are classified and documented following syntax patterns, connecting them with the corresponding LEL symbols. After documentation, the model is verified against the set of scenarios and the LEL.

For the verification process we use a set of questions of the type: a) for a given rule: Which is the context of the organisation in which this rule is applied? What is the associated behaviour? Which are the consequences produced by the application of the rule? b) for a particular scenario: Are there any policies that can modify the behaviour of the episodes?

Finally, the model is validated with customers to detect elicitation errors or organisational conflicts. It is an informal validation with the help of a syntactic-Oriented procedure that identifies subsets of related rules given by the use of LEL symbols [19].

4 A Business Rules Based Process for XP

In this section we describe the XR process. The first Sub-Section describes the updated version of the original XR proposal [4]. The major modification is the inclusion of the CRC cards, not originally proposed in [4]. The following Sub-Sections detail the integration of business rules, the LEL, scenarios and CRCs cards.

4.1 XR for XP

As we mentioned in Section 2, XP carries out its requirements definition process through the continuous construction of stories, tests, new stories and so on, where the client has an active participation to define the scope and priorities of each release. XR is a family of processes based on client-Guided requirements models, easy to build and to understand, that facilitates communication with clients without altering the basic philosophy of XP. Although it generates more documentation, this is easy to obtain and maintain. XR tries to solve some problems that XP presents related to Requirements Process [4] and summarised below:

Problem 1:	The assumption that, in the planning game, business could be represented by just one customer
Problem 2:	The lack of consideration of non-Functional requirements from the standpoint of business
Problem 3:	The lack of explicit links between stories, tasks cards and CRCs to the code
Problem 4:	The lack of a process to produce functional tests
Problem 5:	The lack of a process to produce stories, tasks and CRCs

XR proposes to mitigate these problems with a simplified requirements process based on LEL and scenarios. In this work, we extend XR with the incorporation of the business rules model. Business rules guide the elaboration of simplified versions of the scenarios, and help priority determination for each release. The rules are also key to evolution, since, as the organisation changes, rules will change and thus the changes will propagate to scenarios and CRC cards. Following the XP philosophy, XR simplifies the requirements models construction process. Despite this, the XR process improves the construction of the models in an organised way, resolving the problems pointed above.

Considering the XP iteration philosophy through the releases, XR proposes the construction of business rules and the LEL as a prior activity since the business rules describe an organisation entirely independent of the existence of a software system. We follow the general idea that rules do not imply any technical implementation since they are statements that define or constraint some aspect of the business [20]. Scenarios depend on each release, they can be defined based on the selected rules, as the client can determine priorities and decide to define only scenarios related to a particular set of business rules. Based on the selected scenarios, the CRCs are defined for each release.

4.2 Business Rules and LEL Construction

The strategies for building the LEL and the business rule model, briefly described in Section 3, have been simplified in order to work in the context of XP.

As business rule model discovery and validation require knowledge of the business processes, this model may be obtained starting with interviews or collaborative workshops with customers following the idea presented in [21]. Customers usually express organisational aspects in terms of *limits, rights* and *responsibilities* of the parts involved in the *UofD*. What responsibilities do the actors of the system have? What limits and rights? Why is an activity carried out in a certain way? Does external legislation affect the business directly or indirectly?

In the XP spirit, it is not necessary to follow the writing patterns strictly, instead, engineers and business customers collaborate to create a business rule template that is expressed in natural language, based on common sense and which is directly relevant to the business customer[9]. It is useful to identify which the symbol responsible for each rule is and the other involved terms. Generally, for each functional rule there will be several associate non-Functional rules. Starting from the group of rules defined by the customer, the engineer completes them by carrying out questions that help the customer to raise aspects that have not been clear.

In these interviews or collaborative workshops, characteristic terms of the *UofD* denoting a resource, a process, a person or a sector are identified. LEL is constructed to define the vocabulary, to eliminate any type of ambiguity or wrong usage of the terms.

For each symbol, the notions and behavioural responses are defined and the term is classified as Subject, Object, Verbal Phrase or State. The LEL, at the same time as it facilitates communication with the customer, is also used as a simple mechanism for traceability , since it relates all the models obtained in this and later stages by the use of a restricted language. Therefore, XR implements traceability by means of common terms (LEL) and by the name spacing process that enforces scenarios and CRCs to use LEL symbols, offering a solution for *problem 3* described in Section 4.1. The name spacing process aims to guarantee that the names in the lexicon are preserved in the stories and will be the same names used in the code. That is, the *coding standards* must have a explicit rule on naming the operators and operands with the same names present in the lexicon. In the context of XP, we do not need a complete LEL to start with, but just a list of the important symbols, since the definitions will be added from the continuous feedback of the XP process.

4.3 Building Scenarios Using Business Rules

We propose the use of scenarios instead of User-Stories. Scenario is a description technique that enhances understandability of task related descriptions and communicability among stakeholders. Our scenario description language is very easy to learn and easy to use. In order to integrate our description language to XP, some minor adaptations were necessary. One of them was to focus scenarios not on the actual situation of the *UofD*, but on the desired situation with the presence of the software. The XP task cards may be taken from sub-scenarios.

The use of scenarios allows representing non-Functional aspects, solving *problem 2* described in 4.1. The *Constraint* component of the scenario language allows engineers to represent non-Functional aspects. To mitigate *problem 4* , we propose to write derived situations from the scenario, with the goal of making the system fail.

Related to *problem 5*, we propose the strategy for producing scenarios, briefly described in Section 3.2. Because of the XP spirit, heuristics to derive the scenarios are simplified and we incorporate the use of the business rules model throughout the construction process in order to reflect the organisational policies. In XR, the focus is on getting a very first cut of scenarios, such that they can be a reasonable basis for producing the functional tests and guiding the CRC design process.

The Derive process is concerned with eliciting the information and is performed by Business using the LEL. We use the business rules model to select a set of scenarios to derive CRCs for each release. Business rules are the organisational policies, therefore it is easy for the customer to identify priorities in this model: what functionality does he prefer to implement in each release? Customers select the most representative business rules they want to implement and through the trace mechanism it is possible to identify related scenarios. The Describe process details the scenarios as well as the functional test cases. We add the use of business rules to complete the description. As scenarios are described from one point of view, business rules allow them to represent general behaviour related to the situation but not derived directly with the *on-Site* customer who is used for this particular scenario description. In this way, we mitigate *problem 1.* The Validate activity is performed by Business through the process of sharing viewpoints or viewpoint resolution [22], also mitigating *problem 1.* Verify will be performed by the XP team during the activities of *testing* and *listening*. Note that the Validate sub-Process is not the XP practice *testing*, here we are just making sure that the scenarios and the functional tests express the overall viewpoint of Business.

4.4 CRCs Construction Process

Taking into account the XP spirit, we propose a meeting for CRCs construction. But we also propose a simple construction process to build them, a very simplified version of [19]. We derived CRCs from LEL and Scenarios through a set of heuristics. Since business rules were considered in the scenario construction, CRC cards will model the organisational policies.

Subject LEL Symbols are Candidates to be Modeled as CRCs:

We have to consider Subject symbols that appear in the selected scenarios of each release. When considering limits of the software, we have two options depending on what our automation policy is:

- If the Subject represents an important abstraction for the future system (taking into account related business rules), the symbol becomes a CRC card representing the behaviour the subject performs in the real world or representing a record of this behaviour. The responsibilities of each CRC are defined taking into account the behavioural responses of the corresponding LEL symbol.

- If the symbol ought to be an external entity, then its behavioural responses become system input/outputs managed by other classes. In order to do this, it is analysed who the receptor/generator of the data is. It is important to use scenarios to understand the moment and the precise form of communication of the system with the external entity.

Analyse LEL Symbols that Appear in Previous CRCs Responsibilities (Collaborators):

A list with the terms representing objects, verbal phrases and states that appear in the previous responsibilities is built. These new classes are collaborators of the previous ones. Each one of these symbols is analysed to determine if they will be modelled as a class.

The *Object* is modelled as a class if it has significant behaviour for the system, since it defines an important behaviour that can be modelled as a necessary independent abstraction to collaborate with other classes. To discard it as a class it should be analysed if the impacts of the symbols are not describing similar actions of some primitive class, that is to say that although it is an abstraction characteristic of the system, in terms of objects, it does not justify to be a new class. If the symbol is a *Verbal Phrase, it* can be modelled as A class or A responsibility. It is represented as a class when it has characteristics that cannot be assigned to other classes. If this is the case, it is modelled as a class, otherwise, it becomes one or more responsibilities of the involved classes. Finally, we have to treat the symbols of type state. Each symbol will be modelled as a class if it is not possible to include it on an already existing class, that is, if the state has impacts that can be modelled as independent responsibilities.

For each class, the responsibilities are defined by analysing the impacts of the corresponding LEL symbol. It is necessary to keep in mind the category of the symbol since, as it is the category, it changes the sense of the impacts being able to, in some cases, add responsibilities to other classes.

Modelling Business Rules as CRCs:

There are some cases in which it is possible to model a group of rules as classes, to isolate policies in a single object or because the rules affect different objects and it is not possible to associate them in a particular class. Business rules as classes improve the understanding of the policies that govern the system SINCE they are described explicitly as separate classes and they are not absorbed in any other class of the system. They also improve the maintenance of the system since, if a rule changes, it is easy to propagate the change into the associate rules encapsulated in the same class. Also, it can be specialised independently from the classes it affects. Finally, it is possible to have an activation mechanism in the same object to reflect the dynamism of the rules without affecting the involved conceptual objects [18]. The disadvantage of modelling business rules as classes is that, by adding more classes and their relationships, the complexity of the system increases.

Validate CRC Model through the Scenarios:

The resulting CRCs are evaluated as proposed in [23]. Each scenario is “executed” determining if the involved CRCs and their responsibilities can carry out the actions.

The responsibilities can be refined adding the necessary functionality to satisfy each scenario. As scenarios have been completed with business rules, the resulting CRCs will contain all the functionality defined by the rules.

5 A Case Study: Potatoes

Don Juan is a small agricultural business with 17 employees. In this example we show some aspects of potatoes commercialisation. The elicitation was conducted by interviews with the commercial area. In this Section, we show some examples of the models generated by the strategy. [5] Interviews for business rules elicitation were conducted by simple questions: Who are Don Juan clients? Are all types of clients the same? Is there any price policy ?

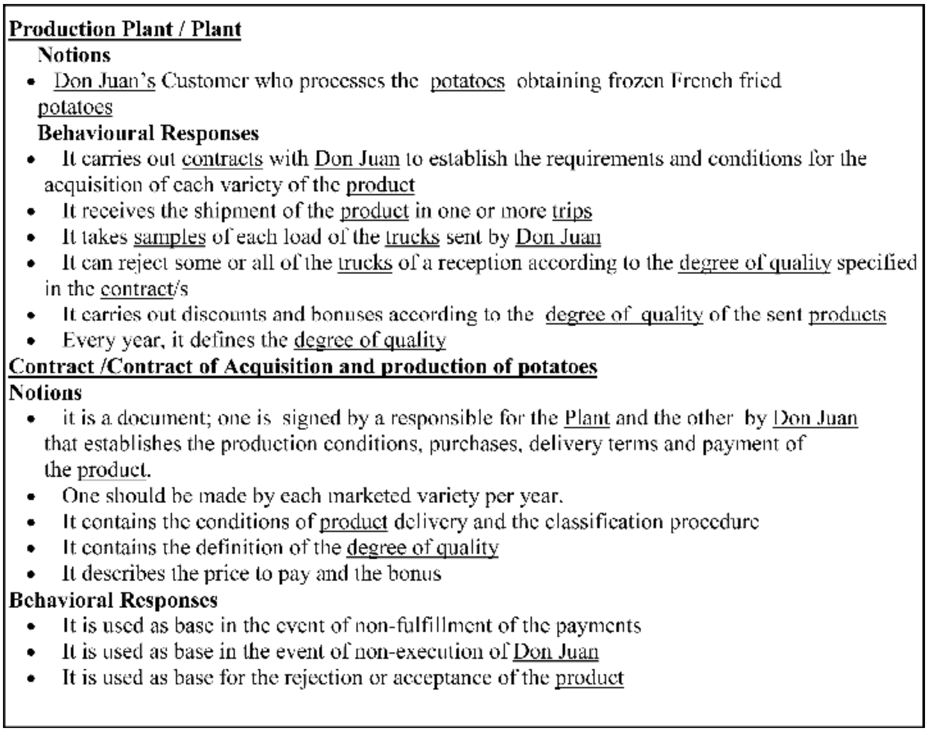
Some examples of business rules elicited from Don Juan:

1. Don Juan sells to Wholesalers, Consignees and Production Plants (FR)
2. The Plant carries out discounts and allowances on the price settled down in the contract starting from the result of the samples. (NFR)
3. The degree of quality is defined by the Plant in the contract (FR)
4. If the Plant rejects trucks, Don Juan fulfils the contract with merchandise bought outside or own. (NFR)
5. The Plant accepts or rejects the product according to the degree of quality obtained in the sample. (NFR)
6. Don Juan uses the result of the samples, carried out by the Plant, to make decisions, as for that of potato production and for selection, to carry out in his next shipment. (RF)
7. According to the financial state, the product stock or convenience of the market, Don Juan sells to Wholesalers or Consignees, choosing them for their commercial characteristics (NFR)
8. When a demand is done, it is analysed if it suits to satisfy it for market prices and for commercial characteristics.
9. If it is not possible to satisfy several clients' demands, it is prioritised for commercial characteristics (NFR)

LEL symbols (underlined in the above rules) were defined during the interviews. Figure 4 shows some examples of the defined symbols.

Taking into account the business rules, we select, for the first release, all the scenarios related to the Plant, the most important client for Don Juan (see rules 1-6). Figure 5 shows an example of a scenario built from Product Delivery point of view (from Behavioural Responses of Don Juan's LEL symbol). During the Describe activity, this scenario was completed according to the business rules model to reflect the delivery aspects with the commercial aspects: episode 4 appears as a consequence of Rule 2 and episode 5 is included to reflect rule 4. In this way, we mitigate problem 1 of Section 4.1

[5] Please, note that all the examples were written in Spanish and were *freely* translated to English



Production Plant / Plant

Notions

- Don Juan's Customer who processes the potatoes obtaining frozen French fried potatoes

Behavioural Responses

- It carries out contracts with Don Juan to establish the requirements and conditions for the acquisition of each variety of the product
- It receives the shipment of the product in one or more trips
- It takes samples of each load of the trucks sent by Don Juan
- It can reject some or all of the trucks of a reception according to the degree of quality specified in the contract/s
- It carries out discounts and bonuses according to the degree of quality of the sent products
- Every year, it defines the degree of quality

Contract /Contract of Acquisition and production of potatoes

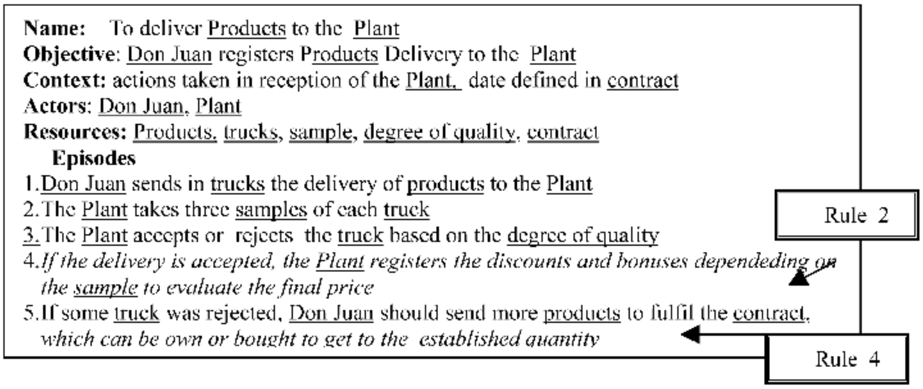
Notions

- it is a document; one is signed by a responsible for the Plant and the other by Don Juan that establishes the production conditions, purchases, delivery terms and payment of the product.
- One should be made by each marketed variety per year.
- It contains the conditions of product delivery and the classification procedure
- It contains the definition of the degree of quality
- It describes the price to pay and the bonus

Behavioral Responses

- It is used as base in the event of non-fulfillment of the payments
- It is used as base in the event of non-execution of Don Juan
- It is used as base for the rejection or acceptance of the product

Fig.4. LEL examples



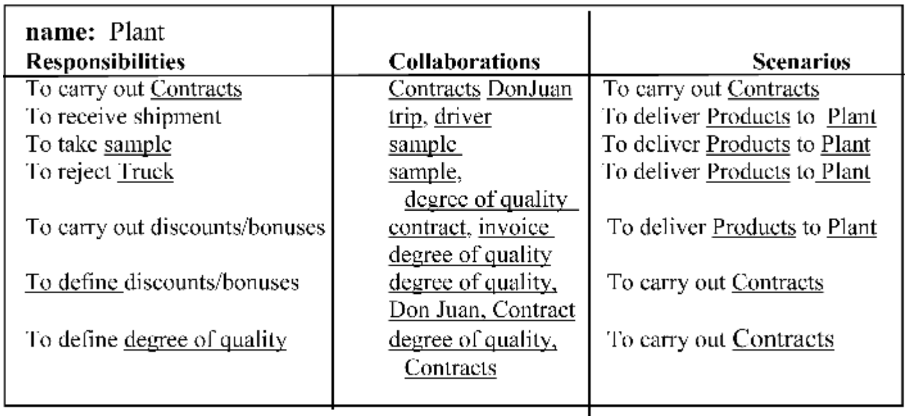
Name: To deliver Products to the Plant
Objective: Don Juan registers Products Delivery to the Plant
Context: actions taken in reception of the Plant, date defined in contract
Actors: Don Juan, Plant
Resources: Products, trucks, sample, degree of quality, contract

Episodes

1.Don Juan sends in trucks the delivery of products to the Plant
2.The Plant takes three samples of each truck
3.The Plant accepts or rejects the truck based on the degree of quality
4.*If the delivery is accepted, the Plant registers the discounts and bonuses dependeding on the sample to evaluate the final price*
5.If some truck was rejected, Don Juan should send more products to fulfil the contract, *which can be own or bought to get to the established quantity*

Fig. 5. Example of Scenario

Finally, CRCs are built taking into account the selected scenarios. Figure 6 shows an example derived from a Subject LEL. The use of scenarios and underlined terms representing LEL symbols enhance traceability.

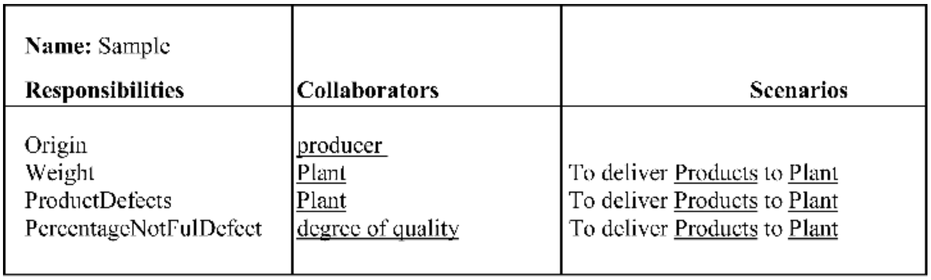


name: Plant

Responsibilities	Collaborations	Scenarios
To carry out Contracts	Contracts DonJuan	To carry out Contracts
To receive shipment	trip, driver	To deliver Products to Plant
To take sample	sample	To deliver Products to Plant
To reject Truck	sample, degree of quality	To deliver Products to Plant
To carry out discounts/bonuses	contract, invoice degree of quality	To deliver Products to Plant
To define discounts/bonuses	degree of quality, Don Juan, Contract	To carry out Contracts
To define degree of quality	degree of quality, Contracts	To carry out Contracts

Fig.6. CRC card of the Plant

The following CRC corresponds to a Sample. Although the sample is carried out by the Plant, it is necessary to define a class that reflects it, since, as defined in rule 6, it is Don Juan's interest to control each sample. It was also suggested that they should register samples of their production and of the one bought to the producers to fulfil the requirements and to do a statistical analysis through different seasons.



Name: Sample

Responsibilities	Collaborators	Scenarios
Origin	producer	
Weight	Plant	To deliver Products to Plant
ProductDefects	Plant	To deliver Products to Plant
PercentageNotFulDefect	degree of quality	To deliver Products to Plant

Fig.7. Sample CRC

The example shows how the business rules, LEL, scenarios and the CRCs cards interact, stressing the traceability aspect addressed by our proposal. Although we did not conduct a full case study, in which a running system is produced, we believe the demonstration provided is sufficient to show a reasonable support of our ideas. Future work is progressing in the direction of a full case study. The client-Oriented nature of the proposed process facilitates and encourages the participation of the customer, in this case, the actor in charge of Don Juan's commercial sector. Each CRC is derived from LEL and scenarios, keeping the trace to its origin, and since scenarios were constructed following the business rules model, we can assert that the CRC model reflects the organisational policies.

6 Conclusion

Our proposal integrates well with the XP philosophy. Requirements evolution [3] has been a major concern in our research and we see XP as a possible paradigm to transfer to practice our evolution ideas. As such, we propose in this paper a first step in this direction that is how to substantiate the XP process with a requirements focus based on business rules. We do not discuss the applicability of XP, nor its successes or failures. We have just analysed the requirements part and we have proposed an add-on which we do believe has high cohesion with the XP proposal.

XP does not propose an explicit requirement strategy, it presents several problems in its informal way of dealing with requirements, as we described in Section 4.1. We consider that the use of natural language oriented requirements models fits well in the spirit of XP, since they allow engineers to represent the organisation through a set of models that are easily validated with customers. As Kotonya and Sommerville suggest in [24] and Sommerville et Sawyer describe in Rule 8.8. *"Paraphrase System Models"* [25] is fundamental to have a natural language oriented model to allow the participation of customers.

Our strategy favours the adaptative, customer-Oriented and incremental vision of XP, in particular: LEL unifies the vocabulary allowing communication with the customer. Scenarios improve User-Stories techniques through their representation, construction and inspection process. Business rules describe the knowledge from a simple and close way of the form in which customers perceive the organisation. They also guide in the releases definition by allowing the client to identify his priorities from this declarative model. Since there are no explicit heuristics to derive CRC, we think that our proposal shows how to derive CRC cards directly from the requirements models enhancing traceability and reflecting the real requirements and organizacional policies. The requirements models are adaptable to changes: A change in the organisation can affect the policies, invalidating or generating new business rules. The customer identifies the changes easily and how they impact in the rules, since these are defined in his own language.

On the other hand, some of the controls we have defined in [13] are not used in favour of an agile development. Of course it will impact the overall quality of the scenarios, but here we sustain, without data, that the results reported by the XP community are a strong argument to relax these controls. Due to the increasing popularity of this kind of methods [26], known as agile[www.agilemodeling.com], and the explosion of customer-Oriented Internet applications, which may need different production techniques[27], we believe that the requirement engineering area has to research and propose simple requirements models and strategies to be used in this context. Our proposal to this challenge is based on our experience in requirements, and in the idea that natural oriented requirements models fit in with the XP philosophy. Experience on the integration of XR is essential. We hope to have the opportunity to implement these ideas and gather data on their use. We also see this proposal as a way of exposing to the object oriented community what the role of requirements in a programming centered paradigm is.

References

1. Beck, K., Extreme Programming Explained Embrace Change, Addison Wesley Longman, Inc., (2000)
2. Fowler Martin “New Methodology”, http://martinfowler.com/articles/newMethodology.html
3. Leite, J.C.S.P., Scenario Evolution. Dagstuhl-Seminar-Report; 199, Schloss Dagstuhl, Internationales Begegnungs-und Forschungszentrum Fur Informatik, Bui, Carrol and Jarke (editors), Alemanha, (1998) 13-14
4. Leite J.C.S.P “Extreme Requirements (XR)”, Jornadas de Ingeniería de Requisitos Aplicada. Sevilla, 11 y 12 de Junio (2001)
5. Beck K., Cunningham W, “A Laboratory For Teaching Object-Oriented Thinking:” From the OOPSLA'89 Conference Proceedings October 1-6 (1989) 1-6
6. Leite, J.C.S.P, Rossi, G., Maiorana, V., Balaguer, F., Kaplan, G., Hadad, G., Oliversos, A. Enhancing a Requirements Baseline with Scenarios, Proceedings of the Third International Symposium on Requirements Engineering, IEEE Computer Society Press (1997) 44-53
7. Leite, J.C.S.P., Anchoring the Requirements Process on Vocabulary, Requirements Capture, Documentation and Validation, Dagstuhl Seminar Report – 241 (1999). www.dagstuhl.de/DATA/Reports/99241
8. Leite J.C.S.P, Leonardi Ma. Carmen, “Business rules as organizational Policies”, IEEE IWSSD9: Ninth International Workshop on Software Specification and Design, IEEE Computer Society Press (1998) 68-76
9. Gottesdiener “Business RULES Show, Power, Promise”, Application Development Trends, Vol 4, nro. 3 (1997) http://www.ebgconsulting.com/
10. Leite J.C.S.P, Oliveira A. Pádua Albuquerque. “A Customer Oriented Requirements Baseline”, Proceedings of the Second IEEE International Symposium on Requirements Engineering , IEEE Computer Society Press (1995) 108-115
11. R.Wirfs-Brock, B. Wilkerson, and L. Wiener Designing Object-Oriented Software, Prentice Hall International, Englewood Cliffs, NJ, (1990)
12. Coleman et al., Object-Oriented Development The Fusion Method, Prentice Hall, Englewood Cliffs, NJ, (1994
13. Leite J, Hadad G, Doorn J, Kaplan G., “A Scenario Construction Process” Requirements Engineering Journal, Springer-Verlag. Vol.5 N1 (2000) 38-61
14. Leonardi Carmen, Leite J.C.S.P, Rossi G., "Estrategias para la identificación de Reglas de Negocio", Proceeding de Sbes98 "Simposio Brasilero de Engenharia de Software" Sociedad Brasilera de Computacao, Brasil, 14-16 de Octubre (1998) 53-67
15. Schmauch, Ch., ISO 9000 for software Developers, revised Edition, ASQC, Quality Press (1995)
16. Fiorini, S., Leite, J.C.S.P., Macedo-Soares, T., "Integrando Processos de Negocio a Elicitacao de Requisitos" Revista de Informática Teorica e Aplicada, Instituto de Informática da Universidade Federal do Rio Grande do Sul, Vol. IV, N. I. 7-48

17. Yu E., Modelling Strategic Relationships for Process Reingeneering, PhD Thesis, University of Toronto (1995)
18. Diaz, O., Iturrioz, J., Piattine, M., “Promoting business policies in object-oriented methods” Sesión Trabajos ya publicados: publicado en The Journal of Systems and Software, 1998. Actas de II Jornadas de Ingeniería de Software, JIS97, Dpto. de Informática , Universidad del país Vasco, San Sebastián, España (1997) 384-400
19. Leonardi Carmen, Una Estrategia de Modelado Conceptual de Objetos basada en Modelos de Requisitos en Lenguaje Natural. Tesis de Magister en Ingeniería de Software, Dpto. de Informática de la Universidad Nacional de La Plata (2001)
20. Guide Business Rules Project, “Defining Business Rules - What are they are really”, (1996) www.guide.org/pubs.htm
21. Gottesdiener Ellen “Business Rules as Requirements” Software Development. Volume 7, No. 12. December (1999) http://www.ebgconsulting.com/
22. Leite, J.C.S.P, Freeman, P. A. “Requirements Validation Through Viewpoint Resolution” IEEE Transactions on Software Engineering: Vol. 17, N. 1 (1991) 1253 - 1269
23. Cockburn Alistair “Responsibility-based Modeling”, Technical Memo HaT TR-99.02. http://members/aol.com/humansandt/techniques/responsabilities.htm.
24. Kotonya G, Sommerville I., Requirements Engineering, J. Wiley and Sons, (1998)
25. Sommerville I, Sawyer P “Requirements Engineering: A good practice guide” J. Wiley and Sons (1997)
26. Highsmith Jim and Cockburn Alistair “ Agile Software Development: The Business of Innovation”. IEEE Computer, V.24, N.8. 120-122
27. Brokat Advisor TM “Ruling a Self-Service World” http://www.brokat.com/brwhitepapers/advisor_selfservice.pdf