# Quality Control and ISO Quality Compliance in the Product Lifecycle Management at Siemens

CLOTILDE ROHLEDER Centre de Recherche en Informatique Université Paris 1 Panthéon Sorbonne, F-75013 Paris Siemens PL (DE) GmbH, D-90475 Nuremberg GERMANY clotilde.rohleder@siemens.com

*Abstract:* - From our experience with customers who deploy customized software products, we have learned that deriving products from shared software assets requires more than complying with quality standards like ISO9126. Additionally, developers must consider what we call the quality profile of the final product. A process that matches the quality profile of final product during product derivation helps provide and validate industrial software application solutions. This paper describes this matching concept and its application in a case study of the development of a product lifecycle reporting tool at a large organization. Also we propose a tool to improve compliance with ISO Quality Standards and that could enhance quality control in derivation process.

*Key-Words:* - Quality Product Derivation, Requirements Engineering, Non-Functional Requirements, ISO Quality Standards, Quality Compliance

# **1** Introduction

Developing any system, even one for a single customer, requires addressing the customer's functional and non-functional requirements (Quality of Service). However, many developers lack a convenient way to address the non-functional requirements. Especially difficult is handling variability of non-functional requirement.

Our project addresses this shortcoming by considering derivation process of a product starting from the quality requirements of a customer. We base our research on quality standards like ISO9126 International The Organization for [13]. Standardization (ISO) classifies the Non-functional Requirements by building a Software Quality tree (Fig. 2). In addition to referring to IS09126 standards, we consider the different customer priorities relative to the industry domain to build the quality profile of the final product. The impact of Non Functional Requirements on Variants may vary [23]. We treat the variability from a functional and non-functional requirement (NFR) perspective [17] and [9] [10] to derive products.

The remainder of this article is organized as follows. The next section describes the state of the art. In Section 3, we describe our approach of quality product derivation. Section 4 explains the context of Product Lifecycle Management and summarizes a case study that applied our approach. In Section 5, we propose an implementation of our approach as a quality control tool and evaluate the contribution of our approach for the Product Lifecycle Management in Section 6. We conclude and mention future research work in Section 7.

# 2 State of the Art

Prior derivation approaches like Deestra's approach [5] or like RED-PL approach [6] focuses on the decision-making process during product configuration. For example at some point in time a software customer must choose initial requirements, that involves selecting some and excluding some undesired requirements. We extend the work of Djebbi and Salinesi [6], [7] that considers also the product family. Quality profile and matching process is not found in current derivation approaches. The domain scope denotes the extent of the domain or domains in which the product family is applied, and consists of four levels of scope, i.e. single product family, program of product families, hierarchical product families [2] and product population [19]. The concept of domain knowledge for derivation is in accordance with the RED-PL approach [6] [7] and its corresponding CL language that uses both requirement attributes, and the domain of the attributes. Our approach considers the identification of selection criteria as mentioned in [18] in order to select product. We consider NFRs as attributes [14]. We use the extended notation of [4],

[8], [15], [16], and [17] to get the NFRs representation. [1] [2] and [25] concentrate on implementation aspects of system variability and do not consider the non-functional requirements. Our approach differs from approaches in traditional models like Halmans and Pohl's [11] that describe variability with use case diagrams and place dependences in a separate model. But like [11], we

represent all variability types and cardinalities that are associated to variants [23].

Our Quality Product Derivation covers all important scope of Software Requirements engineering: requirements elicitation, Analysis, Requirements Traceability, and Validation according to [12].



Fig. 1: Quality Product Derivation Process

# **3** Quality Product Derivation

We will first give an overview of our approach of Quality Product Derivation. Then we will explain the Quality Profile of Final Product and the matching process using similarity metrics.

# 3.1 Overview of Quality Product Derivation

Fig. 1 shows the four steps of the Quality Product Derivation: define the quality profile of final product, build the derived product using the matching process, realize the derived products and validate and test the derived products [21]. In each step, we have to consider information about the quality requirements and the NFR impact on Variants [22] [23]. Each step is iterative. The quality product derivation is the construction of a software product that is built by matching the selection of product family artifacts structure with the quality profile of the final product. We will focus in this paper on the first two steps.

# 3.2 Quality Profile of Final Product

The first step of the Quality Derivation Process defines the quality profile of the final product. This quality profile addresses different goals from the different customer points of view. Many important goals are "non-functional". They are not the same goals according to the sector they belong to. Table 1 shows the quality profile of final product of the reporting tool for three companies of different sectors: Defense, Entertainment and Manufacturing. ISIC Category D, F and I are in the domain scope.

Table 1. NFR Goals from the two customer points of view

Automotive	Entertainment
Easy migration and	High availability: Reuse
portability:	components, use standard
Compatible interfaces	components and
for OEM and end	protocols,
user, multi CAD	build rugged hardware
systems on different	components,
platform	improve diagnostic
infrastructures	services and update
(Windows, SUN,	services, web-based
IBM, HP)	
Innovation: Choose	High scalability: Pay
innovative platforms,	attention to the low-end
use up-to-date	solutions.
interfaces, Build a	Add performance as
system with a future	desired.
High flexibility: Open	<u>High usability:</u> Analyze
architecture,	important use cases and
distribution of	optimize interaction, work
functions possible	towards consistent,
	intuitive tool interfaces

Most of the product families we encountered in practice have a profile that can be classified according to two dimensions of scope: quality and domain scope. The first dimension, quality scope, is arranged according to NFR classification based on ISO9126 (Fig. 2). It captures almost all common and different characteristics of the product family members (configurable product family).



Fig. 2: Software Quality according to ISO 9126

In addition to the quality scope, we identify a second dimension, the domain scope. The domain scope denotes the extent of the domains in which the product family is applied [2] [19]. We use the standard classification of economic activities published by the United Nations Statistics Division: International Standard Industrial Classification of All Economic Activities for studying the different qualities expectations according to the industry category they belong to. The Structure Level 1 is list of tabulation categories marked by one-letter alpha code- A to Q. For our research work, we will focus on the 4<sup>th</sup> till 11<sup>th</sup> category as shown in Table 2. Scalability is provided through all ISIC sub-sections.

No.	Letter	Industries
	Code	
4	D	Manufacturing
5	Е	Electricity, gas and water supply
6	F	Construction
7	G	Wholesale and retail trade; repair of motor
		motorcycles and personal and household go
8	Н	Hotels, restaurants, tourism
9	Ι	Transport, storage and communications
10	J	Financial intermediation
11	K	Real estate, renting and business activities

**Table 2.** ISIC Categories for Quality Product Derivation.

The textual representation of Quality Profile of Final Product is a compilation of the NFRs according to a formula to compile all ISO9126 NFRs. The formula (1) is applied to compile all NFRs and their coefficients. Notation: QPP: Quality Product Profile:

 $QPP_{ISIC-lass} = \sum_{i=1}^{n} NFR_i.SatisficingValue_i$ 

(Formula 1) Formula for Quality Profile of Final Product

Explanation of the algorithm principle:

We set QPP<sub>class</sub> the name of Quality Product Profile for a specified product classification. The values of NFR<sub>i</sub> according to the ISIC classification: e.g. NFR *Operability*, NFR *Usability* We recorded three values for SatisficingValue<sub>i:</sub> {++, +,0}. The NFR can be quantified according to the three first Chung's satisficing values [4]: very satisficed (++), partly satificed (+), neutral (0). *n* is the index of the defined NFRs within ISO-9126-NFRs.

As an example the quality profile of final product for Automotive (G) concerning a PLM module could be: QPP<sub>iso9126-ISIC-G</sub>=Portability[PLM-module].++. It means the NFR goal of companies belonging to Automotive Industry (OEM suppliers) addresses in particular the portability which has high importance as the satisficing value expected is "very satisficed" (++).

Fig. 3 shows an example of graphical representation of Quality Profile of Final Product for ISIC Entertainment and Product Report



Fig. 3: Example of Quality Profile of Final Product for ISIC Entertainment and Product Report.

According to our industrial research in the PLM field, we could find the following quality profile for final product: Automotive (G), Defense (sub category of I), Entertainment (sub category of H) and Manufacturing (D).

## **3.3 Matching Process according to the Quality Profile of Final Product**

In this step the derived product is going to be built according to the defined quality product profiles. The matching process consists in going through all the variant combinations found in the description basis of product line and select the combination matching the quality product profile of final product. The map model as quality product model provides the structure of variants according to their quality attributes. The variants analysis considers the quality profile of the final product to select some variants and delete others. Through the matching process one can obtain a variant structure among all variants combinations of the subset of shared product family assets. The final structure is the derived product.

#### 3.3.1 Variants

As explained in [22], [23], we use the model of presenting NFR's impact on variant based on Map model [20] to represent Variants including NFR impact. Map is a process model expressed in a goal driven perspective. It provides а system representation based on a non-deterministic ordering of goals and strategies. The map is represented as a labeled directed graph (see an example in Fig. 4 with goals (Goal) as nodes and strategies (Strategy) as edges between goals and Section as ways to achieve the target goal from the source goal. The directed nature of the graph shows which goals can follow which one. For example (see Fig. 8) Strategy  $S_{ii}$ between the couple of goals  $G_i$  and  $G_j$  represents the way  $G_i$  can be achieved once  $G_i$  has been satisfied. Section  $\langle G_i, G_i, S_{ii} \rangle$  represents a way to achieve the target goal  $G_i$  from the source goal  $G_i$  following the strategy  $S_{ii}$ . We also use a textual notation in which a section named  $ab_i$  designates a way to achieve a target goal b from a source one a following a strategy *i*. Thus, the section  $\langle G_i, G_j, S_{ij} \rangle$  is named  $ab_1$  where a is the code of the goal G<sub>i</sub>, b is the code of the goal  $G_i$  and *1* is the code of the strategy  $S_{ii}$ .



As explained in [23] and [3] the features represented in a map are related to each other by four kinds of relationships, namely *multi-thread*, *bundle*, *path* and *multi-path* relationships. These relationships show the possible combination of features from which the user can select the appropriate ones according to user needs. We map these combinations of features to variants.

A variant is a representation at requirements level of a cohesive bundle of system functionalities according the user's point of view. We define different variant types corresponding to the different relationship types inside the map: atomic, simple and composite variant.

Atomic variants are not decomposable into other variants. They are linked directly to system functionalities. The atomic variants are linked with each other to build variants with bigger granularity (simple or composite).

**Simple variants** consist of atomic variants. These can be exclusive (linked by a *link of type: alternative choice*). The atomic variants can be complementary (linked by a *multiple choice link*) when some of them can be selected. According the choice link type, the simple variants are specialized into: *simple variants with alternate choice and simple variants with multiple choice*.

**Composite variants** are aggregates of variants linked by a *composition link*. The composite variants are specialized into various sub types according the composition link type between their components: *path composite variant* and *multi-path composite variant*. Each possible functionalities composition builds a path composite variant. The bundle of possible compositions builds a multi-path composite variant.

The NFR Impact on Variant refers to the ISO 9126. We have to first select NFRs. Then we have to quantify NFRs value. We apply a value of impact of NFR [4] on variant/MAP section as explained in [22] and [23]. This is called the quality attribute of the variant (QoS). As an example for suitability, we have to quantify the execution of instructions and function blocks, the transfer of data and time response. In the case of interoperability, we have to quantify the correct interchange of data via specified bus systems (CPU<->peripheral units, bus master <->slaves). One more example addresses the reliability compliance, we have to quantify how the test object fulfills the requirements of standards and internal temperature, requirements regarding climate, etc: Quantifying Non Functional vibration, Requirements is done according to Chung's NFR satisficing value representation: This value is "satisficed" when the customers' expectation is met (++). The value is called partly satificed (+) if the customer annoyance is reached and if there is the risk of no customer acceptance (0). Finally the value is called partly not satisficed (-) and absolutely not satisficed (--) in case of customer refusal if there is concern that the customer will not buy the product. We emphasize that it is important to specify exact values of satisficing values at later points in time for each NFR. Fig. 5 shows that the impact of NFR Performance on variant V is the value in sec. or min. or MB, etc. The impact of NFR Performance on Variant V concerning a PLM-module is written: QoS(V)=Performance[PLM-module].++.



### **3.3.2** Similarity metrics

We adapt the deep semantic relations in the initial coefficients formula of Dice (Formula 3) to get the *Modified Coefficient of Dice* [24]. The formula (2) corresponds to the modified weighted Dice's coefficient, where A is the NFR impact on a variant (QoS<sub>V</sub>) and B is the quality profile of final product (QPP<sub>ISO9126-ISIC</sub>). The coefficients  $\alpha_{ij}$  define the weight granted to the similarity between the different NFRs occurring in A and B. The sum of coefficients  $\alpha_{ij}$  is equal 1.

$$\mathcal{S}_{D}^{\boldsymbol{w}}(\boldsymbol{A},\boldsymbol{B}) = \sum_{i,j} a_{ij} \times \mathcal{S}_{D}^{\boldsymbol{m}}(\boldsymbol{A}_{i},\boldsymbol{B}_{j})$$

(Formula 2) Weighted Modified Coefficient

Metrics defined by the modified coefficients of Dice are used for computing the measures explained here after. Similarity metrics are applied on A and B. Intrinsic Factor and of Criteria Synonymy). The synonymy is a relation between A and B having properties whose names or values have a deep semantical similarity. We three different similarity degrees to evaluate if A and B are synonymous:

(1) IDE metric when the value of A is *identical* to the value of B, if they are expressed in the same terms and if they have exactly the same meaning, IDE(A,B)= true if  $S^m_D(A,B)$ =1.

(2) SIM metric when the value of A is *alike or similar* to the value of B, if these two values are identified with different terms but if they have the same meaning,

 $SIM(A,B) = true if 0,90 = < S^{m}_{D}(A,B) < 1.$ 

(3) CLOSE metric: when the value of A is *close* to the value of B when these two values are expressed with different terms, but have close meaning,  $CLOSE(A,B) = true \text{ if } 0.80 = <S^{m}_{D}(A,B) < 0.90.$ 

The second metric is called **TIH** (Types of similarity of Intrinsic Factor and of criteria Hyponymy/Hyperonymy). If A and B are linked by a relationship *Is-A*, they can be hyponymous or hyperonymous when the sense of one extends / includes the sense of the other one. The applied metric for hyponymy is HYPO. HYPO(A,B)= true if  $0,60 = <S_{D}^{m}(A,B) < 0,80$ . The applied metric for hyperonymy is HYPER. HYPER(A,B)= true if  $0,40 = <S_{D}^{m}(A,B) < 0,60$ .



# 4 Case Study

## 4.1 Product Lifecycle Management (PLM) Context

Product lifecycle management (PLM) is "the process of managing the entire lifecycle of a product from its conception, through design and manufacture, to service and disposal. PLM integrates people, data, processes and business systems and provides a product information backbone for companies and their extended enterprise". Fig. 6 shows that PLM is a strategic business approach that applies a consistent set of business solution in support of collaboration creation, management, dissemination, and use of product definition information across the extended enterprise from concept to end of life.



Fig. 6: Overview PLM Product Lifecycle Management

Installing a PLM system implies –like with other complex COTS such as ERPs - some kind of matching between users' requirements and the requirements that the system is able to satisfy [20]. The worldwide PLM applications market in 2007 amounted to \$ 8.7 billion. There is no doubt that PLM is now broadly valued by large manufacturers as well as by small and medium-sized business. Cost, effort, time and complexity of implementing a PLM system can be compared to those for implementing an ERP system. Requirements Engineering in PLM includes the storing and managing of requirements. Quality control is focused on how to respond to the customer requirements in the PLM system. Product lifecycle management is the process of managing the entire lifecycle of a product from its conception, through design and manufacture, to service and disposal. Fig. 7 shows the integration of non-functional requirements concept within PLM.



Fig. 7 Integration of Non-Functional Requirements within PLM

The main purpose of a PLM system is to enable collaboration among users. PLM systems handle a large collection of collaborative data (requirements specifications, simulation data, design 2D files, 3D models, bill of material, production plans, sales and marketing data, logistics, etc) from the early stage of product development until the maintenance phase. As a result, PLM systems have extremely diverse kinds of users: requirements engineers, CAD designers, CAE and CAM engineers, ERP users, maintenance technicians, etc. Each have specific expectations with respect to the PLM tool, not only in terms of functionality, but also in terms of ergonomy, performance, interoperability with other systems, and ability to support business goals. Besides, PLM systems must handle extremely different fields of application which results in extremely different NFR priorities [21] [22] [23]. For example in OEM Automotive Supplier sector, NFR "Portability" has "absolute" priority because of multi CAD systems (NX, ProE, CATIA, etc.) on different platform infrastructures (Windows, SUN, IBM, HP) used by key PLM users, whereas in the field of Defense the NFR "Security" comes first.

Most of the time, each requirement has to be translated into the PLM system language and model. We have to translate customer requirements into product quality attributes, and decompose and transform product quality attributes into component quality attributes, part quality attributes, geometric feature quality attributes and tolerance quality attributes. PLM support tools are expected to handle both hardware and software development. The need to provide a seamless workflow from design to manufacturing phases has forced PLM systems to handle not only the documents produced, but also much of their internal contents (metadata) as well. A detailed information model of the product data is an integral part of a PLM system.

#### 4.2 Case Study

To validate our approach we conducted a case study at Siemens PL that extended a prior study [23]. It considers the reporting tool for Product Lifecycle Management. Fig. 8 represents the Functional Requirements that the system must fulfill to provide a data reporting tool for Product Lifecycle Management.

The reporting tools provides the PLM user a report concerning the Bill-of-Material, where-used data, the where referenced data with respect to the access rules of the PLM user, its group and role. Such a tool has an important role in Product Lifecycle Management. It reports collaborative data (cooperation between various stakeholders in a multi-site location context). The purpose is to manage productivity (e.g. change management), documentation (e.g. bill of material), adopted commercial off-the-shelf and customized components, quality and risk.

Our solution consists in representing the variants of PLM reporting Tool including Quality Attributes and to present the adequate derived product of the product family for three companies from different industrial sector. On this PLM reporting tool, we have applied our approach of Quality Derivation for three different companies. This also considers the quality profile of final product and matching process.

# 4.2.1 Variants

We define nine atomic variants. Fig. 8 is composed of 2 goals "*Identify report*" and "*Conceptualize report*" to create a report.



Fig. 8: Functional Requirements of a PLM Data Report Tool

## 4.2.2 NFR impact on Variants (QoS):

We used the Non-Functional Requirements Performance, Security and Informativeness whose decomposed subgoals are Time[ProduceReportStatement], Confirmation [PLMDataForReport] and SecurityWorkflowData [ProduceWorkflowReport] [23]. Table 3 lists the NFRs impact on atomic variants of Fig. 8.

Table 3: NFR impact on atomic variants

	A
Name	Value of NFR Impact on Variants
ab <sub>1</sub>	ab <sub>1</sub> .Time[ProduceReportStatement].+,
	ab <sub>1</sub> .Confirmation[PLMDataForReport]
$ab_2$	ab2.Time[ProduceReportStatement],
	ab2.SecurityWorkflowData[ProdWorkflData].+
$bb_1$	bb <sub>1</sub> .Time[ProduceReportStatement],
	bb <sub>1</sub> .SecurityWorkflowData[ProdWorkflData]
$bc_1$	bc <sub>1</sub> .Time[ProduceReportStatement].+,
	bc <sub>1</sub> . SecurityWorkflowData[ProdWorkflData].?
$bc_2$	bc2.SecurityWorkflowData[ProdWorkflData].+
$cc_1$	cc1.Time[ProduceReportStatement],
	cc1.Confirmation[PLMDataForReport].++
$cc_2$	cc <sub>2</sub> .Time[ProduceReportStatement],
	cc2.Confirmation[PLMDataForReport].++
$cb_1$	cb1.Confirmation[PLMDataForReport]
$cd_1$	cd <sub>1</sub> .Time[ProduceReportStatement].++,
	cd1.Confirmation[PLMDataForReport].+

#### 4.2.3 Quality Profile of Final Product

Table 4 shows the quality profile of final product of the reporting tool for three companies of different sectors: Defense, Entertainment and Manufacturing. ISIC Category D, F and I are in the domain scope.

Table 4. Quality Profile of Final Product for three Companies.	
----------------------------------------------------------------	--

Manufacturing	Defense	Entertainment
(ISIC <sub>D</sub> )	(ISIC <sub>I</sub> )	(ISIC <sub>H</sub> )
PLM Report	PLM Report	PLM Report
Efficiency.++	Efficiency.+	Efficiency.0
Security.+	Security.++	Security.0
Informativeness.0	Informativeness.0	Informativeness.++

#### 4.3. Matching process

According to the matching process, the similarity typologies and the metrics are performed with the formula of modified weighted DICE coefficient (Table 5).

Table 5. Results	of IDE,	SIM,	CLOSE,	HYPO.	HYPERO.

Variants	Results of Matching Process			
	Manufacturing	Defense	Entertainment	
	(ISIC <sub>D</sub> )	(ISIC <sub>I</sub> )	(ISIC <sub>H</sub> )	
$ab_1$	True	True	False	
$ab_2$	False	True	True	
$bb_1$	False	False	False	
$bc_1$	True	True	True	
$bc_2$	True	True	True	
$cb_1$	True	True	True	
$cc_1$	False	False	True	
$cc_2$	False	False	True	
$cd_1$	True	True	True	

9 Fig. shows the results of derived variant/product. Applying our approach, we obtained preliminary design views for the reporting tool that were implemented in the resulting reporting system. By their positive responses, leaders who participated in this case study suggested our approach is worthwhile for wining billable projects. The derived product according to the profile 1 (Defense): ab1 has been accepted, ab2, cc1, cc2 have been denied. We follow the same way to get the derived product according to the product classification profile 2 (Manufacturing) and profile 3 (Entertainment).

For confidentiality reason, we have not been authorized to publish detailed results from the case study. We can say participants were interested in having a quality product derivation way to derive the reporting product for customers with different application domains and priorities. As the results of



case study got a positive response, we have been asked to implement the approach in the PLM

software Teamcenter of Siemens PL (DE) GmbH.

Fig. 9: Results of Matching Process in Quality Product Derivation

# **5** Quality Control Tool in the Product Lifecycle Management

We have implemented our approach as Quality Control tool in Teamcenter. The standard system had to be customized. The Data Model had to be extended to support the issues of representing the Non-Functional Requirements, the linking and also the impact of non functional requirements on variants, the Quality Profile of Final Product based on ISO9126, the matching process required a customization of Standard Workflow and also new libraries for applying the algorithm of matching process in an action handler.

#### **Non-Functional** 5.1 Representing Requirements, ISO9126, Quality Profile of Final Product

Table 6 shows the data model extension for the New Class and Properties for Simple Element NFR\_Requirement.

Database name	Туре	Display name (EN_US
Item_id	Int	Id
Object_name	String	Name
NFR_goal	String	NFR-goal
SatisficingValue	Int	Coefficient
ISO_cat	LOV	ISO 9126-cat.

**Table 6: Data Model Extension for NFR Element** 

The Business Model BMIDE of Teamcenter is XML based and the Data Model extension has been done in business objects.xml

<tcbusinessdata< th=""><th></th></tcbusinessdata<>			
xmlns="http://teamcenter.com/BusinessModel/TcBusines			
sData" Date="" TcVersion	="">		
<add></add>			
<tcstandardtype t<="" td=""><td>ypeName="NFR_Requirement"</td></tcstandardtype>	ypeName="NFR_Requirement"		
parentTypeName="Requir	ement"		
typeClassName="Item"/>			
<tcattribute< td=""><td></td></tcattribute<>			
attributeName="Name"attri	ributeType="POM_long_string		
"/>			
<tcattribute< td=""><td>attributeName="NFR_goal"</td></tcattribute<>	attributeName="NFR_goal"		
attributeType="POM_long	_string" />		
<tcattribute att<="" td=""><td>ributeName="SatisficingValue"</td></tcattribute>	ributeName="SatisficingValue"		
attributeType="POM_int"	/>		

<tcattribute< th=""><th>attributeName="ISO_cat"</th></tcattribute<>	attributeName="ISO_cat"
attributeType="POM_long_stri	ing" />

Our proposed PLM Data Model extension is colored in yellow in Fig. 10. In our approach, we have added Requirements Engineering classes to the class "Element": SpecElement, Requirement, and NFR Requirements.



Fig. 10: PLM Quality Data Model

Fig. 11 shows the results of PLM data model extension for the new class NFR in Teamcenter Unified Architecture.

۱ 😂	leues Element	X		
Sch	hritte 🔚	😑 🕕 Legen Sie grundlegende Informationen für das neue Element fest.		
3 0	Elementinformatione Geben Sie die Element-Mas	Element-ID / Änderungsstand-ID - Name REQ-000040 * / A • SecurityWorkflo* (Zuvelsen)		
8	Grundlegende Kennungsinf	NFR-goal ProduceWorkflowReport		
9 9	Zusätzliche ID-Information Zusätzliche Änderungsstan	ISO 9126 Cat.: Security		
	Zurück	Weiter 🕨 🦓 Beenden 🏠 Startseite 🛞 Schließen		

Fig. 11: SecurityWorkflowReport[ProduceWorkflowReport]

ISO9126 provides Quality Standards that are listed in a List OF Value. The LOV is also an extension of PLM Data Model.



<tclovattach <="" lovname=" ISO_cat_Prop" td=""></tclovattach>
TypeName="ISO_cat">
<tclovattachpropertyinfo< td=""></tclovattachpropertyinfo<>
ValuePropertyName="ISO_cat_Prop" />

Quality Standards ISO9126 can be implemented using the new class and grouping the standard quality NFRs in a container Requirements Manager. Fig. 12 shows the implementation of ISO9126 and quality profile of final product for Automotive ISIC G, defense (sub cat. J), Entertainment (sub c. of H).



Fig. 12: ISO9126 and Quality Profile of Final Product

# 5.2 Representing Variants and SatisficingLink

We represent the variants and SatisficingLink as NFR impact on variants. Products Structures are represented through the PSE Product Structure Editor as shown in Fig. 13.

DSF - Teamcer	nter 2007			
Datal Dasibaltas	Analytic Underson Provider UKs			
Datei Bearbeiten	Ansicht werkzeuge Fenster Hilfe			
🗵   😂   🔜	% 6 6   ⊞ ≘ ¥   → ** i*   *	1 🛠 🖉 🎽	0 0 3 8 8 6	JGS Teamcenter
🕲 Zurück 🔹 🎯 🔸	PSE (infodba (infodba) - dba/DBA [IMC-49884891	5])		🧾 ×
	Erste Schritte Mein Teamcenter 📝 *PSE 🗙			- 0
Geben :	000019/A;1-VCp(a{bc}d) - Latest Working - Datum -	"Heute"		III 🔛
Qu <b>Aripaislen 🔺</b>	Elementbeschreibung	Elementname	Stücklisteneintrag	Elementtyp
Startseite	Path Composite Variant	VCp(a(bc)d)	000019/A;1-VCp(a{bc}d) (vi	Item
Meine Arbeitski	😑 🤧 Atomic Variant	VCp(a{b}c)	000011/A;1-VCp(a{b}c) (view)	Item
Meine gesneich	😑 🤣 Simple Variant with Alternate Choice	VSa(ab)	000006/A;1-VSa(ab) (view)	Item
Maine Links		ab1	000004/A;1-ab1	Item
W MELLE LINS		ab2	000005/A;1-ab2	Rem
		bb1	000010/A;1-bb1	Item
	😑 🤭 Simple Variant with Multiple Choice	V5m(bc)	000009/A;1-VSm(bc) (view)	Item
et 1 100 a		bc1	000007/A;1-bc1	Item
🎾 Erste Schri…		bc2	000008/A;1-bc2	Item
-	😑 🤣 Path Composite Variant	VCm(c{b}c)	000017/A;1-VCm(c{b}c) (vi	Item
🌐 Mein Team	😑 🤭 Atomic Variant	VCp(c{b}c)	000013/A;1-VCp(c{b}c) (view)	Item
- DCC	Simple Variant with Multiple Choice	V5m(bc)	000009/A;1-V5m(bc) (view)	Item
27 PSC		bc1	000007/A;1-bc1	Item
😹 Schedule		bc2	000008/A;1-bc2	Item
1921		cb1	000012/A;1-cb1	Item
🛐 CAE SE		bb1	000010/A;1-bb1	Item
	😑 🤭 Path Composite Variant	VCp(cc)	000016/A;1-VCp(cc) (view)	Item
V Prüfung		cc1	000014/A;1-cc1	Item
6 Chaudhann		cc2	000015/A;1-cc2	Item
m structure		cd1	000018/A;1-cd1	Item
👍 DesignCon	<			>
🖻 🗄 🌖 »		- 💽	2 × % d*	19 🏁 🔮 😂 😂
📓 Bereit				<u></u>

Fig. 13: Variant Representation in Teamcenter Unified Architecture

Data Model has to be extended here to include a new Compound Property QoS and a new linking between NFRs and Variant. In Teamcenter, the linking of customer functional and non-functional requirements is accomplished by defining quality relations among the associated quality attributes to the product to be managed. Variants are represented as Product Structures through the Product Structure Editor tool (PSE). PLM Data Model has been extended here to enable the new linking between NFRs and Variant.

The linking between the NFRs and the Product elements called variant in our model can be done through extending the Data Model with a new relation called "Satisficing Link". This is made in Business\_object.xml. Here is an extract of the file content.

<tcstandardtype< th=""><th>typeName="SatisficingLink"</th></tcstandardtype<>	typeName="SatisficingLink"			
parentTypeName="TC_Link"				
typeClassName="TC_Link"/>				
rules.xml				
<tcgrmrule <="" primarytypename="NFR_Requirement" td=""></tcgrmrule>				
secondaryTypeName="Variant"				
relationTypeName="SatisficingLink"				
primaryCardinality="0"	secondaryCardinality="0"			
secured="false" attachability	="WriteAccessReq"			
changeability="Changeable"				
detachability="WriteAccessReq"/>				

Fig. 14 shows the results of PLM data model extension for the new relation "Satisficing Link"between PrimaryObject NFR and secundaryObject variants in Teamcenter.



Fig. 14: Representation of Variant and NFR Impact on Variant

#### **5.3 Matching Process for Product Derivation**

A new workflow "Match-ISO9126-Compliance-ISIC-K" has been created (Fig. 15) to apply the formula of modified weighted DICE coefficient.

Workflow Designer (infodba (infodba) - dba/DBA [IMC-498848915])					
Erste Schritte Mein Teamcenter PSE 🔁 w	orkflow Designer 🗙				
Prozessvorlage Match-ISO9126-Compliance-	ISIC-K	Vorlagentyp   Prozess	51		
Match-ISO9126-Compliance-ISIC-K	Starten	Match-ISO9126-Compliance-ISIC-K	_		

Fig. 15: Process "Match-ISO9126-Compliance-ISIC"

The Data Model of Teamcenter Unified Architecture has to be extended with new status. Here is an extract of the content of the file options.xml.

xml version="1.0" encoding="UTF-8"?
<tcbusinessdata< th=""></tcbusinessdata<>
xmlns=http://teamcenter.com/BusinessModel/TcB
usinessData Date="" TcVersion="">
<add></add>
<tcstatus description="" statusname=" ISO9126-ISIC-K-&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;compliance-ok"></tcstatus>

When the action handler returns the value ok, the workflow is finished. It means, the status is set and the user gets a dialog: "ISO 9126 Compliance ISIC-K matched"

If the WF returns ok, the variant is referenced in the class view: ISIC-Derived Product (Fig. 16).

🤌 Mein Teamcenter - Teamcenter 2007 📃 🗖 🗙				
Datei Bearbeiten Ansicht Werkzeuge Fenster Hilfre				
🛛   🖻 🕭	📎 🤌 📸 🕺 🖏 🖎 👌 🔍 Suche 🛅 Ordner 📴 🔤 UGS Teamcenter			
😌 Zurück 🔹 😔 🔸 Mein Teamcenter (infodba (infodba) - dba/D&A [IMC-438848915]) 😝 X				
	Erste Schritte 🜐 Mein Teamcenter 🗙 PSE Workflow Designer Requirements Manager 📃 🗆			
📝 PSE	🍐 Persönliche Ablage 🛛 🕹			
😹 Schedule	Crdner × 🛱 ISO9126-ISIC-Derived 🛷 Wewer 🔻			
🛐 CAE SE				
🗹 Prüfung	Complete Brock stilling i			
🗣 Structur	General			
🁍 DesignCo	Anwenden			
🖪 🚟 »	┣<>			
🚚 Bereit				

Fig. 16: Derived Products as Separate View in Teamcenter

# **6** Evaluation of the Approach

#### 6.1 Evaluation of Case Study

Our approach of Quality Product Derivation has been used within Teamcenter Unified Architecture to visualize, clarify non-functional requirements, to navigate through the variants in order to get information about the quality attributes, and to adequately derive product according to customer's requirements. The scalability of the representation is enabled by 1) the decomposition method of the map for the functional requirements and 2) the typology of NFR according to the Chung NFR types. Applying our approach, we obtained preliminary design views for the reporting tool that were implemented in the resulting reporting system. By their positive responses, leaders who participated in this case study suggested our approach is worthwhile for wining billable projects as this can be used as quality assurance. For confidentiality reason, we have not been authorized to publish detailed results from the case study.

# 6.2 Lessons-learned and benefits of the use of the tool

□ The implementation of model is easy to read and understand by the users of the vendors and the system technical stakeholders.

□ The abstraction and decomposition mechanism is useful and possible with the tool during the functional and non-functional requirements modeling. This make it possible to model the requirements globally and to detail the requirements only if this is necessary.

The implementation of hierarchy of requirements represented by variants and of nonfunctional requirements NFRs in the tool make it possible to better structure the requirements analysis. The variants are built and represented in the tool using different level. This enables to communicate the requirements step by step. If we need a general direction, we have to navigate up to the highest level and we can show a general overview of the project requirements. The deep variants level are operative variants and are an effective way to communicate details for some project users.

The tool enables a better traceability of the matching process.

□ The workflow based similarity analysis in an effective way to compare systematically some topics of a big number of models.

The workflow based matching process and building process of derived product avoids the subjective evaluations of consultants.

The tool helps building the final derived product and gives the customer a better decision process.

# 6.3 Contribution for Product Lifecycle Management

Linking Quality Components:

PLM should enable the linking of customers functional and non-functional Requirements on product elements (that we call variants). Our approach suggests using a new Link called "Satisficing Link" found in Chung's NFR Framework [2]. This link is the relation between Variants and Non-Functional Requirements. From the design phase to the manufacturing phase, the information concerning the quality of assemblies' components remains visible to PLM users according to the Access Control Rules. This information remains traceable and retrievable in all steps of life Development. Engineering. cvcle: Design. Manufacturing, Sales, After-Sales, Maintenance, Revisions, Change.

# Version Management:

In PLM systems, revisions of an object are manually managed by the user and form a sequential series, with no possibility of performing parallel changes. There is only a possibility to perform parallel development, if the release status management is implemented to perform branching of product development and merging of the two developed stand. If the quality attributes of one element do not meet the customer's requirements, the versioning process should be applied on the element which gets new quality attributes. The version management of simple parts and of complete assemblies is a challenge won by PLM systems. Complexity is due to the different kinds of modeling items that may exist in a model compared to the single type that are conventionally handled. The handling of quality requirements that are linked with the versioning concept enables quality assurance within PLM projects.

# Quality Configured Product Structure:

In PLM, quality control should address the physical structure of the final product because it is the predominant structure. Geometric features and their assemblies build product structure. This structure is used throughout the development phases as a basis for the information model to which all other information is related. In the model-based approach, PLM focuses on the internal structures of the models stored in the CAD files instead. When using models throughout the development phases, the software structure varies widely, and hence the product structure management functionality of a model needs to handle many different parallel structures. From these product structures one can get the bill of material. The product structure and also the resulting bill of material (BOM) should be able to be configured according to quality.

#### Quality Process Support:

Process Support including workflow management, group and user assignment, approach rule mechanisms is one of the fundamental PLM mechanisms. Our approach of integrating nonfunctional requirements enables to create a new release process called the ISO 9126 Quality Control Process. The ISO9126 quality control process can be applied on simple parts or complete assemblies. The target of this quality control process is to simplify and support the release management and also to integrate a Quality Assurance within PLM concept.

## 7 Conclusion

This paper proposes a Quality Product Derivation using a matching process on the quality profile of final product and NFR impact on variants. To identify the impact of non-functional requirements on variants, we represent the non-functional requirements by goals. We capture the variability through a goal-driven modeling formalism called map. In our Quality Derivation approach, we investigate how the NFR impact on Variants has to be considered in the whole quality product derivation process. To illustrate, we report a case study concerning a Product Lifecycle Management (PLM) reporting tool and we validate our approach in implementing a first version of ISO9126 quality control solution in the PLM software Teamcenter. As future research work, we will focus on extending our approach of monitoring the quality key factors to all quality industry standards within PLM. Our quality data model approach is developed from IEEE standard for software documentation; we refer to ISO 9126 but we should address all PLM final products that include also pure hardware-product and not software components like embedded software. So the variants should be Geometric features relative to 3D models or derived 2D models and Quality Profile of Final Products should include all corresponding relevant ISO Quality Standards.

**Acknowledgments:** We thank Siemens PL (DE) GmbH, Germany for supporting the project.

#### References:

- [1] Bachmann F., and Bass L.: *Managing variability in software architecture*. ACM Press, NY, USA, 2001.
- [2] Bosch J., Florijn G., Greefhorst D., Kuusela J., Obbink H., Pohl K.: Variability issues in Software Product Lines 4<sup>th</sup> International Workshop on Product Family Engineering (PEE-4), Bilbao, Spain, 2001
- [3] Bennasri S., Souveyet C.: Capturing requirements variability into components", 6th International

Conference on Enterprise Inf. Systems ICEIS'04, Porto, Portugal, 2004.

- [4] Chung L., and Subramanian N.: Process Oriented Metrics for Software Architecture Adaptability, *Proceedings of ISRE*, 2001
- [5] Deelstra S., Sinnema M., Bosch J.: Product Derivation in Software Product Families; A case Study, *Journal of Systems and Software*, Vol 74/2 pp. 173-194, January 2005.
- [6] Djebbi, O. and Salinesi S.: RED-PL, a Method for Deriving Product Requirements from a Product Line Requirements Model, International Conference on Advanced information Systems Engineering (CAISE'07), Springer Verlag, Norway, June 2007
- [7] Djebbi O., Salinesi C., and Diaz D.: Deriving Product Line Requirements: the RED-PL Guidance Approach, Asian Pacific Software Engineering Conference (APSEC), IEEE Computer Society, Nagoya, Japan, 2007.
- [8] González-Baixauli B., Sampaio do Prado Leite J.C., and Mylopoulos J.: Visual Variability Analysis for Goal Models. *Requirements Engineering Conf. 2004*: 198-207
- [9] González-Baixauli B., Laguna M.A., Sampaio do Prado Leite J.C.: A Meta-model to Support Visual Variability Analysis, *First International Workshop on* Variability Modelling of Software-intensive Systems, January 2007, Limerick, Ireland.
- [10] González-Baixauli B., Laguna M.A., Sampaio do Prado Leite J.C.: Using Goal-Models to Analyze Variability, Second International Workshop on Variability Modelling of Software-intensive Systems, January 2008, Essen, Germany
- [11] Halmans G., Pohl K.: *Communicating the variability* of a software product family to customers, Software and System Modeling, Springer-Verlag 2003.
- [12] Hashim K., Khairuddin, N., A Framework for Software Requirements Engineering, 8th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems (SEPADS '09), February 2009, Cambridge, UK, pp.117-120.
- [13] ISO/IEC 9126-1: Software Engineering Product Quality - Part 1: Quality Model, 2001
- [14] Keller R.K., Schauer R.: Design Components: Towards Software Composition at the Design Level, *Proceedings of International Conference on Software Engineering*, April 19-25, 1998, Japan, pp. 302-311.
- [15] Lapouchnian A., Y. Yu, Mylopoulos J., Liaskos S., Sampaio do Prado Leite J.C.: From stakeholder goals to high-variability software design, *ftp.cs.toronto.edu/csrg-technical-reports/509*. Tech. rep., University of Toronto, 2005
- [16] Lapouchnian A., Yu Y., and Mylopoulos J.: Requirements-driven design and configuration management of business processes, *BPM* (2007), pp. 246–261.
- [17] Liaskos S., Yu Y., Lapouchnian A., Mylopoulos J., and Sampaio do Prado Leite J.C., "From Goals to High-Variability Software Design", *ISMIS* 2008.

- [18] Muhairat I. M., Al-Kasasbeh, B., Al-Qutais, R. E., The Analytical Hierarchy process as a Tool to select Open Source Software, 8th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems (SEPADS '09), February 2009, Cambridge, UK, pp.39-44.
- [19] Van Ommering, R.; Bosch, J.: Widening the Scope of Software Product Lines – From Variation to Composition. Proceeding of the Second International Conference on Software Product Lines.(SPLC2). August 2002, pp. 328-347.
- [20] Rolland C., Prakash N.: On the Adequate Modeling of Business Process Families, *BPMDS'07 in conjunction with CAiSE'07*, Norway, 2007
- [21] C.Rohleder, "Quality Product Derivation: A Case Study for Quality Control at Siemens", Best Paper Award of 8th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems (SEPADS '09), February 2009, Cambridge, UK, pp. 51-56
- [22] Rohleder, C., "Representing Variants Including Quality Attributes, IPSI Transaction Journal "Transaction on Internet Research, January 2008, Volume 4, Number 1, ISSN 1820 – 4503.
- [23] Rohleder, C.: Visualizing the Impact of Non-Functional Requirements on Variants – A Case Study, International IEEE Conference on Requirements Engineering, REV'08, Barcelona, 2008
- [24] Salinesi, C., Etien, A., Zoukar, I.:A Systematic Approach to Express IS Evolution Requirements Using Gap Modelling and Similarity Modelling Techniques, *CAiSE* 2004.
- [25] Svahnberg M., Van Gurp J., Bosch J.: On the notion of variability in Software Product Lines. *Proceedings* of the IEEE Conf. on Software Architecture, 2001