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 [13]. The International Organization for Standardization (ISO) classifies the Non-functional Requirements by building a Software Quality tree (Fig. 2). In addition to referring to ISO9126 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

<table>
<thead>
<tr>
<th>Automotive</th>
<th>Entertainment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy migration and portability: Compatible interfaces for OEM and end user, multi CAD systems on different platform infrastructures (Windows, SUN, IBM, HP)</td>
<td>High availability: Reuse components, use standard components and protocols, build rugged hardware components, improve diagnostic services and update services, web-based</td>
</tr>
<tr>
<td>Innovation: Choose innovative platforms, use up-to-date interfaces, Build a system with a future High flexibility: Open architecture, distribution of functions possible</td>
<td>High scalability: Pay attention to the low-end solutions, Add performance as desired, High usability: Analyze important use cases and optimize interaction, work towards consistent, intuitive tool interfaces</td>
</tr>
</tbody>
</table>
ISO9126 (Fig. 2). It captures almost all common and different characteristics of the product family members (configurable product family).

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 4th till 11th category as shown in Table 2. Scalability is provided through all ISIC sub-sections.

Table 2. ISIC Categories for Quality Product Derivation.

<table>
<thead>
<tr>
<th>No.</th>
<th>Letter Code</th>
<th>Industries</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>D</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>5</td>
<td>E</td>
<td>Electricity, gas and water supply</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
<td>Construction</td>
</tr>
<tr>
<td>7</td>
<td>G</td>
<td>Wholesale and retail trade; repair of motor motorcycles and personal and household goods</td>
</tr>
<tr>
<td>8</td>
<td>H</td>
<td>Hotels, restaurants, tourism</td>
</tr>
<tr>
<td>9</td>
<td>I</td>
<td>Transport, storage and communications</td>
</tr>
<tr>
<td>10</td>
<td>J</td>
<td>Financial intermediation</td>
</tr>
<tr>
<td>11</td>
<td>K</td>
<td>Real estate, renting and business activities</td>
</tr>
</tbody>
</table>

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:

$$QPP_{\text{ISIC-class}} = \sum_{i=1}^{n} \text{NFR}_i \times \text{SatisficingValue}_i$$  \hspace{1cm} (Formula 1) Formula for Quality Profile of Final Product

Explanation of the algorithm principle:

We set QPP_class the name of Quality Product Profile for a specified product classification. The values of NFR, according to the ISIC classification: e.g. NFR Operability, NFR Usability We recorded three values for SatisficingValue, {++, +, 0}. The NFR can be quantified according to the three first Chung’s satisficing values [4]: very satisfied (++), partly satisfied (+), 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: $QPP_{\text{ISO9126-ISIC-G}}$=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 satisfied” (++)

Fig. 3 shows an example of graphical representation 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 a 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_{ij} between the couple of goals G_i and G_j represents the way G_j can be achieved once G_i has been satisfied. Section <G_i, G_j, S_{ij}> represents a way to achieve the target goal G_j from the source goal G_i following the strategy S_{ij}. 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 <G_i, G_j, S_{ij}> is named ab_i where a is the code of the goal G_i , b is the code of the goal G_j and I is the code of the strategy S_{ij}.

![Fig. 4: MAP example](image)

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 requirements regarding climate, temperature, vibration, etc: Quantifying Non Functional 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 satificed (-) 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\textsubscript{v}) and B is the quality profile of final product (QPP\textsubscript{ISO9126-ISIC}). 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.

\[
S^m_D(A, B) = \sum_{ij} \alpha_{ij} \times S^m_D(A_i, B_j)
\]

(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.

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.
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 ergonomics, 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.

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

<table>
<thead>
<tr>
<th>Variant</th>
<th>Manufacturing (ISIC D)</th>
<th>Defense (ISIC I)</th>
<th>Entertainment (ISIC H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab1</td>
<td>Efficiency.++</td>
<td>Efficiency.+</td>
<td>Efficiency.0</td>
</tr>
<tr>
<td>bb1</td>
<td>Informativeness.0</td>
<td>Informativeness.0</td>
<td>Informativeness.++</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th></th>
<th>Manufacturing (ISIC D)</th>
<th>Defense (ISIC I)</th>
<th>Entertainment (ISIC H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficiency</td>
<td>Efficiency.++</td>
<td>Efficiency.+</td>
<td>Efficiency.0</td>
</tr>
<tr>
<td>Informativeness</td>
<td>Informativeness.0</td>
<td>Informativeness.0</td>
<td>Informativeness.++</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Manufacturing (ISIC D)</th>
<th>Defense (ISIC I)</th>
<th>Entertainment (ISIC H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab1</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>ab2</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>bb1</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>bc1</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>bc2</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>cc1</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>cc2</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>cc3</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>cc4</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

Fig. 9 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 winning billable projects. The derived product according to the profile 1 (Defense): ab1 has been accepted, ab2, bc1, 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.

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.

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.

5.1 Representing Non-Functional 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.

<table>
<thead>
<tr>
<th>Database name</th>
<th>Type</th>
<th>Display name (EN_US)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item_id</td>
<td>Int</td>
<td>Id</td>
</tr>
<tr>
<td>Object_name</td>
<td>String</td>
<td>Name</td>
</tr>
<tr>
<td>NFR_goal</td>
<td>String</td>
<td>NFR-goal</td>
</tr>
<tr>
<td>SatisficingValue</td>
<td>Int</td>
<td>Coefficient</td>
</tr>
<tr>
<td>ISO_cat</td>
<td>LOV</td>
<td>ISO 9126-cat.</td>
</tr>
</tbody>
</table>

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

```xml
<TcBusinessData xmlns="http://teamcenter.com/BusinessModel/TcBusinessData" Date="""" TcVersion=""">
  <Add>
    <TcStandardType typeName="NFR_Requirement" parentTypeName="Requirement" typeClassName="Item"/>
    <TcAttribute attributeName="Name"attributeType="POM_long_string"/>
    <TcAttribute attributeName="NFR_goal"attributeType="POM_long_string"/>
    <TcAttribute attributeName="SatisficingValue"attributeType="POM_int"/>
  </Add>
</TcBusinessData>
```
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.

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

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.
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.

```xml
<TcStandardType
typeName="SatisficingLink"
parentTypeName="TC_Link"
typeClassName="TC_Link"/>
```

Rules.xml

```xml
<TcGRMRule
primaryTypeName="NFR_Requirement"
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 secondaryObject 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.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<TcBusinessData
xmlns=http://teamcenter.com/BusinessModel/TcBusinessData
Date="" TcVersion="">
<Add>
<TcStatus statusName="ISO9126-ISIC-K-compliance-ok" description=""/>
</Add>
</TcBusinessData>
```

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).

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 winning 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 non-functional 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 cycle: Development, Design, Engineering, 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 non-functional 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:


