Evaluating the Importance of Analyzability, Testability and Changeability Quality Attribute in Reference to Software Product Line Orthogonal Variability Model

Satinderjit Kaur Gill
Dept. of CS
Mewar University, Chittorgarh.
Chittorgarh, India

Dr. Amita Sharma
Dept. of CS&IT
The IIS University
Jaipur, India

DOI: 10.23956/ijarcsse/V7I6/0200

Abstract-In Software Engineering, quality evaluation is very difficult task because of complexity of software systems. For this purpose the evaluation of quality attributes is very important concept with reference to Software Product Line Orthogonal Variability Model. Identifying whether the Orthogonal Variability Model is easy to analyze, test and change will help to develop a successful product line. Three important sub characteristics of Maintainability i.e. Analyzability, Testability and Changeability play a greater role in development of successful product line. This paper is intended to study the role of Analyzability, Testability and Changeability quality attributes over Orthogonal Variability Models.

Keywords- quality, Software product line, maintainability, analyzability, testability, changeability, OVM

I. INTRODUCTION

Instead of developing an individual product, the growing trend in Software Engineering is to develop multiple and similar product at the same time. Due to some constraint like time and cost it is not possible for a software developer to develop new software for each new customer. Software Product Line Engineering (SPLE) provides a solution to eliminate this type of problem. Here Line means a set of products those are related and share some commonalities like data structures, software components, some features and architecture etc[9]. Software Product Line (SPL) is a set of software intensive systems that share a common, managed set of features to satisfy the specific needs of a particular market segment or mission that are developed from a common set of core assets in prescribed way”[1]. In software product line development two phases are defined: Domain Engineering and Application Engineering. In domain engineering, the common software artifacts are designed and developed for reuse. In application engineering, the specific products are derived by reusing a set of the aforementioned domain artifacts [2]. With reference to the latest software quality model proposed by ISO(International Standard Organization) i.e. ISO/IEC 9126 model, Maintainability is one of the six characteristics. Other characteristics are Functionality, Efficiency, Portability, Reliability and Usability. All these characteristics have their own sub characteristics [3]. Maintainability is a very important quality attribute and management of this quality attribute is still a problematic area. It has its own sub characteristics like analyzability, testability, changeability etc. Measuring quality at early phase of development is the key area to develop high quality software product line.

This work focus on the assessment of maintainability related to software product line orthogonal variability models, primarily focusing on analyzability, testability and changeability.

Section II contains information about Software product line. Section III has information about orthogonal variability modeling. Section IV hold the information about quality attributes, and section V analyses the available metrics and Section VI has conclusion and future scope of work.

II. SOFTWARE PRODUCT LINE ENGINEERING

Software product line engineering is concerned with capturing the commonalities universal and shared attributes of a set of software intensive applications for a specific problem domain [4]. It allows rapid development of variants of domain specific application through various configurations of a common set of reusable aspects often known as core assets, which support the management of commonality as well as variability [5]. Due to some factors like time and cost it is not possible for developer to develop a new product from scratch for every new customer that’s why software reuse has increased its importance. Software product line engineering is the technique that offers the ways to eliminate this type of problem.

III. ORTHOGONAL VARIABILITY MODEL

Orthogonal Variability model is an important approach that we use for documenting the variability in product line. In OVM, only the variability of product line is documented. In this model, variation point (VP) documents a variable item and a variant (V) documents the possible instances of a variable item [6]. Variation point identifies a particular requirement of a product line that could be different among different products of a same product line. It is denoted by VP. Variation point is represented by a triangle shape in OVM as shown in diagram below:
Variant postulates a requirement variant for a variation point which is available for particular product of product line for making the use of it. It is denoted by V. A variant is represented by rectangle shape in OVM as shown in diagram below:

Constraint dependency is also used to documents different constraints on the selection of different variation points (VP) and variants (V). There are generally two types of constraints used in OVM like Excludes and requires. Require constraints are generally represented by dashed line with single arrow, while excludes constraints are generally represented by dashed line with double arrow. These both constraints are shown between variation points and variants.

The OVM model represents the variation points and different variants of product line which have been shown by diagram of Travel Agency product line.

**IV. SOFTWARE QUALITY ATTRIBUTES**

Software quality is the degree to which software possesses a desired combination of attributes [7]. In our approach, we define the quality attribute as a measurable property of an artifact. We consider only those properties that can be quantified and technically defined [2]. We can classify quality attributes into two categories: internal attribute and external attributes. We can measure the internal attributes on the basis of products features like size, complication, capability etc. On the other hand we can measure external quality attributes on the basis of products maintainability, efficiency, reliability etc.

Maintainability is the parameter concerned with how the system in use can be restored after a failure, while also considering concepts like preventive maintenance. That’s why we can say that maintainability means the capability of the software product to be modified. Modification may include corrections or any improvements of the software according to changes in environment, and in functional specification and requirements. It can be defined in terms of its analysability, changeability, testability and reliability and readability. Analysability characterizes the ability to identify the root cause of a failure within the software. It is the capacity of the model of software product to be analysed for scarcity. Changeability characterizes the amount of effort to change a system. It is the prospect and comfort of change in product
model when modifications are essential. Testability is a measure of how easy it is to create test criteria for the system and its components, and to execute these.

If in a single-systems achieving quality attributes is sometimes a challenge, in software product line this challenge is complicated because there is variability on quality attribute requirements and different quality constraints are required [8]. For any product line to continue to function successfully and evolve as per need, it is imperative to look upon all the quality attributes that may affect in future [10].

V. EXISTING METRICS

A set of metrics proposed by EbrahimBagheri at al.has used following measures which are given in table 1 for assessing the maintainability of software product line feature model.

<table>
<thead>
<tr>
<th>Measure type</th>
<th>Measure name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size measure</td>
<td>Number of Features(NF)</td>
</tr>
<tr>
<td></td>
<td>Number of top features(NTop)</td>
</tr>
<tr>
<td></td>
<td>No of leaf features(NLeaf)</td>
</tr>
<tr>
<td>Structural Complexity</td>
<td>Flexibility of changeability(FoC)</td>
</tr>
<tr>
<td></td>
<td>Ratio of Variability(RoV)</td>
</tr>
<tr>
<td></td>
<td>Cyclomatic complexity(CC)</td>
</tr>
<tr>
<td></td>
<td>Cross Tree Constraints(CTC)</td>
</tr>
<tr>
<td></td>
<td>Coefficient of Connectivity(CoC)</td>
</tr>
</tbody>
</table>

These Metrics are useful for accessing the quality of feature models, but by implementing these metrics on feature model it works on all common and different features. But these metrics need to be further studied by implementing these on OVM then it will give different results because OVM model is related to only different features on to common features.

VI. CONCLUSION AND FUTURE SCOPE

Most of researcher has proved that we can measure the software quality. But most of them accessed the software quality based on feature model but in future more analysis required on Orthogonal Variability models and need to develop more metrics which will work on only different features because OVM is related to only different features it does not support common features.

REFERENCES


