Research Inventy: International Journal Of Engineering And Science
Vol.3, Issue 10 (October 2013), PP 37-46
Issn(e): 2278-4721, Issn(p):2319-6483, Www.Researchinventy.Com

A Generalized Quality Model for Localized Software Product

[1]Manisha Bhatia, [2]Aparna Sharma
[1]*(Assistant Professor, Computer Science, Banasthali Vidyapith, Jaipur)*
[2]*(M.Tech. Scholar, Computer Science, Banasthali Vidyapith, Jaipur)*

ABSTRACT : *There are a number of quality model already available like MC.Call Quality model, Boehm Quality model, ISO 9126, Capability maturity model and LISA(Localisation Industry standard association) has also introduced its own QA model. Each model is adapted by several organizations in some way to manage the quality assurance process for the components in the product.*

The purpose of this paper is to simulate some of the already existing QA models with regard to the localized product and to introduce a new Generalized QA model combining the advantages of several. It also discusses the QA model with an example of the tester module named GUNVATTA PARIKSHAK and identifies the value metric for each of the operation.

KEYWORDS: *Boehm, CMM, Furps, ISO, LISA, Mc-Call*

I. INTRODUCTION

1.1. Quality

Quality is a measurable characteristic or attributes in comparison to the standards. It is a state of being free from defects, deficiencies and significant variations.

Quality of design refers to the characteristics that software designers specify for a product and *Quality of conformance* is the degree to which the design specifications are followed during development.

For a localized product it will be evaluated as:-

User satisfaction = compliant product + good quality + delivery within budget and schedule

1.2. Quality Control

Quality control involves the series of inspections, reviews, and tests used throughout the software process to ensure each work product meets the requirements placed upon it.

1.3. Quality Assurance

The goal of quality assurance is to provide developer with the data necessary to be informed about software quality, thereby gaining insight and confidence that product quality is meeting its goals.

1.4. Software Quality Assurance

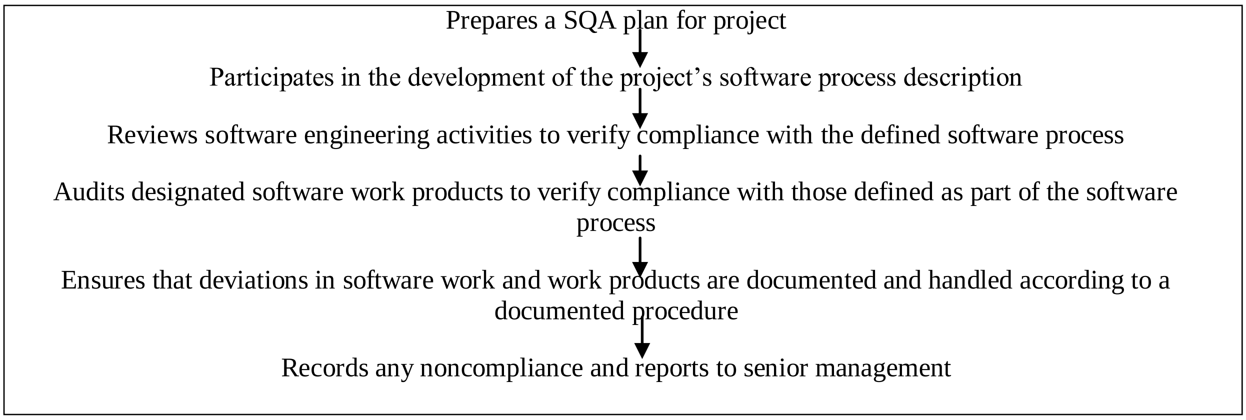
Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software. Wikipedia defines Software Quality assurance (SQA) as “It consists of a means of monitoring the software engineering processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards” but for Localized software product its Quality Assurance verifies that the released localized product can accurately deliver its intended function to its end users. There are a number steps that may be included in QA of a localized product like

[1] Setting up the build environment.
[2] Compiling or building the localized application.
[3] Acquiring the hardware/software and setup needed to run the application.
[4] Acquiring application and test plan learning curve familiarity.
[5] Generating and adopting the formal QA plan with team buy-off.
[6] Implementing the QA plan in runtime according to plan.
[7] Applying any localization changes where necessary, together with any engineering changes required by localization.
[8] Rebuilding the corrected application.

[9] Verifying corrections.
[10] Generating a final QA report

1.5. Software Quality Assurance Activity

Software quality assurance is composed of a variety of tasks associated with two different constituencies—the software engineers who do technical work and an *SQA* group that has responsibility for quality assurance planning, oversight, record keeping, analysis, and reporting. These activities are performed by SQA group that:-



Prepares a SQA plan for project

Participates in the development of the project's software process description

Reviews software engineering activities to verify compliance with the defined software process

Audits designated software work products to verify compliance with those defined as part of the software process

Ensures that deviations in software work and work products are documented and handled according to a documented procedure

Records any noncompliance and reports to senior management

II. EVALUATING KEY PERFORMANCE INDICATORS FOR SOFTWARE EVALUATION

Key performance indicator is a type of measurement in performance to achieve operational goal to achieve zero defects in localized product or 100% customer satisfaction. Performance of a localized product should be measured against goals such as boosting the profitability of international operations, Grabbing share from competitors, improving customer's loyalty, increasing lifetime value of customers and pumping up brand awareness in target regions.

Table 1

Impact of bad localization or not localizing a software product

Marketing	Sales	Support
Limited reach of marketing programs — customers don't know what we're selling	Sales are limited to those who are comfortable in the nuances of the specific language; incomplete transactions	Frustrated customers; bad reputation for usability; brand damage; higher support costs; expensive problem resolution through call centers

III. VARIOUS EXISTING QUALITY ASSURANCE MODELS

3.1.Mc Call Software Quality Model Adopted for Localized Product

It attempts to bridge the gap between the local users and the developers of the localized product. Software quality factors that reflects the user's views and the developer's priorities.

Local Product Revision (ability to undergo changes)

a. Maintainability: - the effort required to locate and fix a fault in the local product .

b. Flexibility:-The ease of making changes required by the changes in the product acc. To the target market and customer.

c. Testability:-Testing the local product based on various quality criteria's and to ensure that it meets its specification and the requirement of the local people

3.2. Local Product Transition (adaptability to the local market)

a. Portability: - Effort required transferring the product from one local market to another with some minor changes.

b. Reusability:-The ease of reusing the software in some different context. For ex. A Localized software product that sells Burger can be used for selling Vada Pav for other region.

c. Interoperability:- The effort required to couple a localized system with other localized system.

3.3. Local Product Operations:- (its operation characteristics)

a. Correctness:-The extent to which a local product fulfills its specification and requirements of local people.

b. Reliability: - The local systems ability not to fail and the information provided must be reliable acc. To the local market.

c. Efficiency:-the use of resources (like processor time, storage etc.) that are required by the local product. May be further categorized as *Execution efficiency and Storage efficiency.*

d. Integrity:-Protection of the Localized product for unauthorized access with passwords, digital signature etc.

e. Usability:-The ease of learning and using the software by the local people.

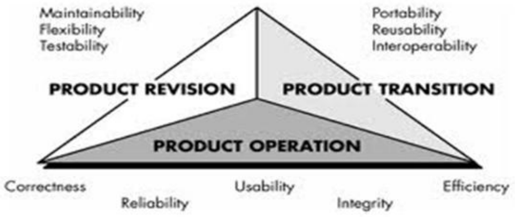


Figure 1: The McCall quality model (a.k.a. McCall's Triangle of Quality) organized around three types of quality characteristics.

All these factors of Mc. Call quality model can be adopted for localized software but with varying degree of acceptance.

Table 2

Quality factors of Mc Call quality model and their priority in requirements when adopted for a localized software product

Quality Factors	**Sub categorization**	**Highly Required**	**Required**	**Required but acceptable with minor changes**
Local Product Revision	Maintainability	☑		
	Flexibility	☑		
	Testability	☑		
Local Product Transition	Portability			☑
	Reusability		☑	
	Interoperability			☑
Local Product Operations	Correctness	☑		
	Reliability			☑
	Efficiency	☑		
	Integrity		☑	
	Usability	☑		

3.4. Boehm Software Quality Model for Localized Product

This model attempts to qualitatively define the software quality by a given set of attributes and metrics. It presents a hierarchical software quality model structured around high level characteristics, intermediate level characteristics, primitive characteristics – each of which contributes to the overall quality level for local products (K.K. Aggarwal and Yogesh Singh 2007).

3.5. High Level Characteristics

a. As-is-utility: - How easily, reliably and efficiently a local product can be used as it is.
b. Maintainability: - How easy a local product is to understand, modify and retest.
c. Portability:- Can a local product can be used if the environment is changed.

3.6. Intermediate Level characteristics

a. Portability:-A local product can be adapted and operated easily on different computer configuration.
b. Reliability:-A local product either performs its expected intended function satisfactorily or not.
c. Efficiency:-A local product is fulfilling its purpose without waste of its resources.
d. Usability:-Ease of use of a local product for its local customer.
e. Testability: - It facilitates the establishment of verification criteria and supports evaluation of its performance.
f. Understandability: - Ease of learning and understanding the purpose of the localized product for its intended local customer.
g. Flexibility:-Incorporation of changes once the nature of the desired changes in the local product is determined.

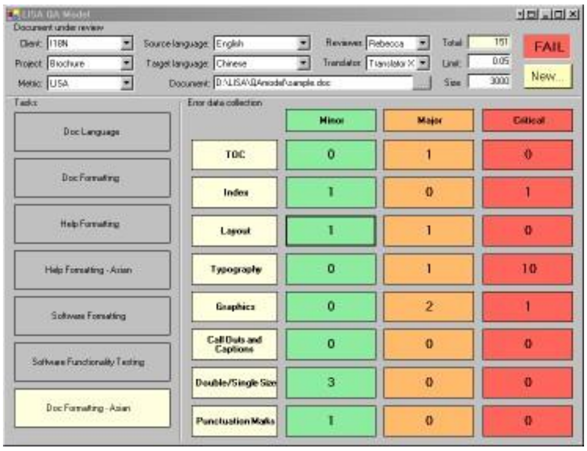
Table 3

Quality factors of Boehm Quality model and their priority in requirements when adopted for a localized software product

Quality Factors	**Sub categorization**	**Highly Required**	**Required**	**Required but acceptable with minor changes**
As is utility	Reliability	☑		
	Efficiency		☑	
	Usability	☑		
Maintainability	Testability	☑		
	Understandability	☑		
	Flexibility	☑		
Portability	-----			☑

3.7. LISA Quality Assurance Model 3.1

The Localized industry Standard Association (LISA) designed a quality assurance model for all the components in a localized product in terms of its functionality, documentation and language issues. It can be implemented as a standalone application and can be easily integrated into any system that supports ODBC database connectivity (LISA QA model 3.1).It provides a number of features to the users of LISA QA model



LISA QA Model

Document under review

Client: I18N
Source language: English
Reviewer: Rebecca
Total: 151

Project: Brochure
Target language: Chinese
Translator: Translator X
Limit: 0.05

Metric: USA
Document: D:\LISA\QAmodel\sample.doc
Size: 3000

FAIL

New...

Tasks:

- Doc Language
- Doc Formatting
- Help Formatting
- Help Formatting - Asian
- Software Formatting
- Software Functionality Testing
- Doc Formatting - Asian

Error data collection

	Minor	Major	Critical
TOC	0	1	0
Index	1	0	1
Layout	1	1	0
Typography	0	1	10
Graphics	0	2	1
Call Outs and Captions	0	0	0
Double/Single Size	3	0	0
Punctuation Marks	1	0	0

Figure 2: Snapshot of quality model given by LISA

1.1.1. It tests all error messages and severity levels in a single click and results in showing the status in PASS/FAIL

1.1.2. Single interface provides access to all like data recording, reporting, adding new projects etc.

1.1.3. It has example data that help users in their implementation after testing the quality of the software.

1.1.4. Metrics for evaluation of the source documents.

1.1.5. QA data can be protected against deletion, alteration etc.

1.1.6. User portable database scheme can be implemented in user own database engine.

1.2. Capability Maturity Model for Localized Product

CMM is a strategy for improving software process, irrespective of the actual life cycle model used. It can be used to judge the maturity of the localized software process and to identify the key practices that are required to increase the maturity of these processes.

CMM maturity model is organized into five maturity levels:-

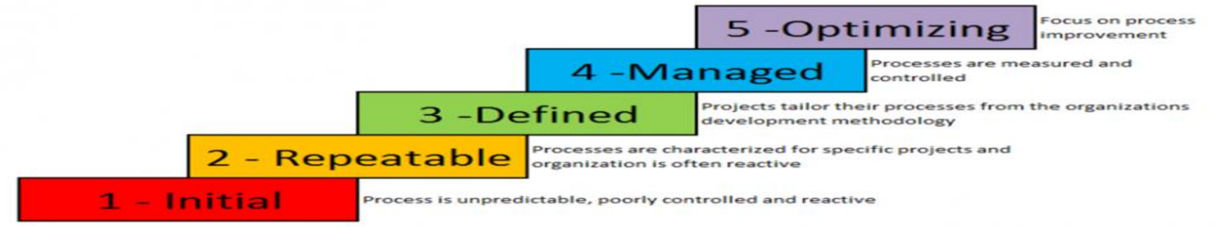

5 -Optimizing
Focus on process improvement
4 -Managed
Processes are measured and controlled
3 -Defined
Projects tailor their processes from the organizations development methodology
2 - Repeatable
Processes are characterized for specific projects and organization is often reactive
1 - Initial
Process is unpredictable, poorly controlled and reactive

Figure 3: Maturity levels of Capability maturity model

Other than all these there are other quality models(like ISO 9126,Furps/FURPS+,Dromey's quality model etc.) each with their advantages and disadvantages.

IV. GENERALIZED QUALITY MODEL FOR LOCALIZED PRODUCT

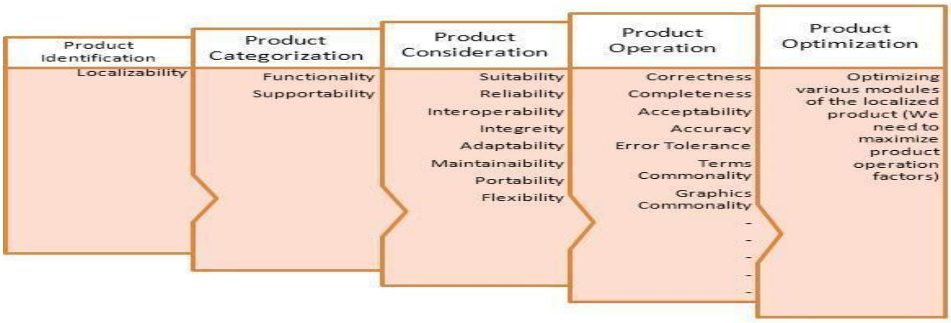


Figure 4: A summarized view of Quality Assurance model for localized product

The Fig 5 represents a complete quality model for localized software combining advantages of various quality models like

1.2.1. It uses the maturity levels of Capability maturity model
1.2.2. It has hierarchical quality model structured around 5 levels i.e.
a. Product Identification
b. Product Categorization
c. Product Consideration
d. Product Operation
e. Product Optimization
1.2.3. It determines both qualitative as well as quantitative factors.

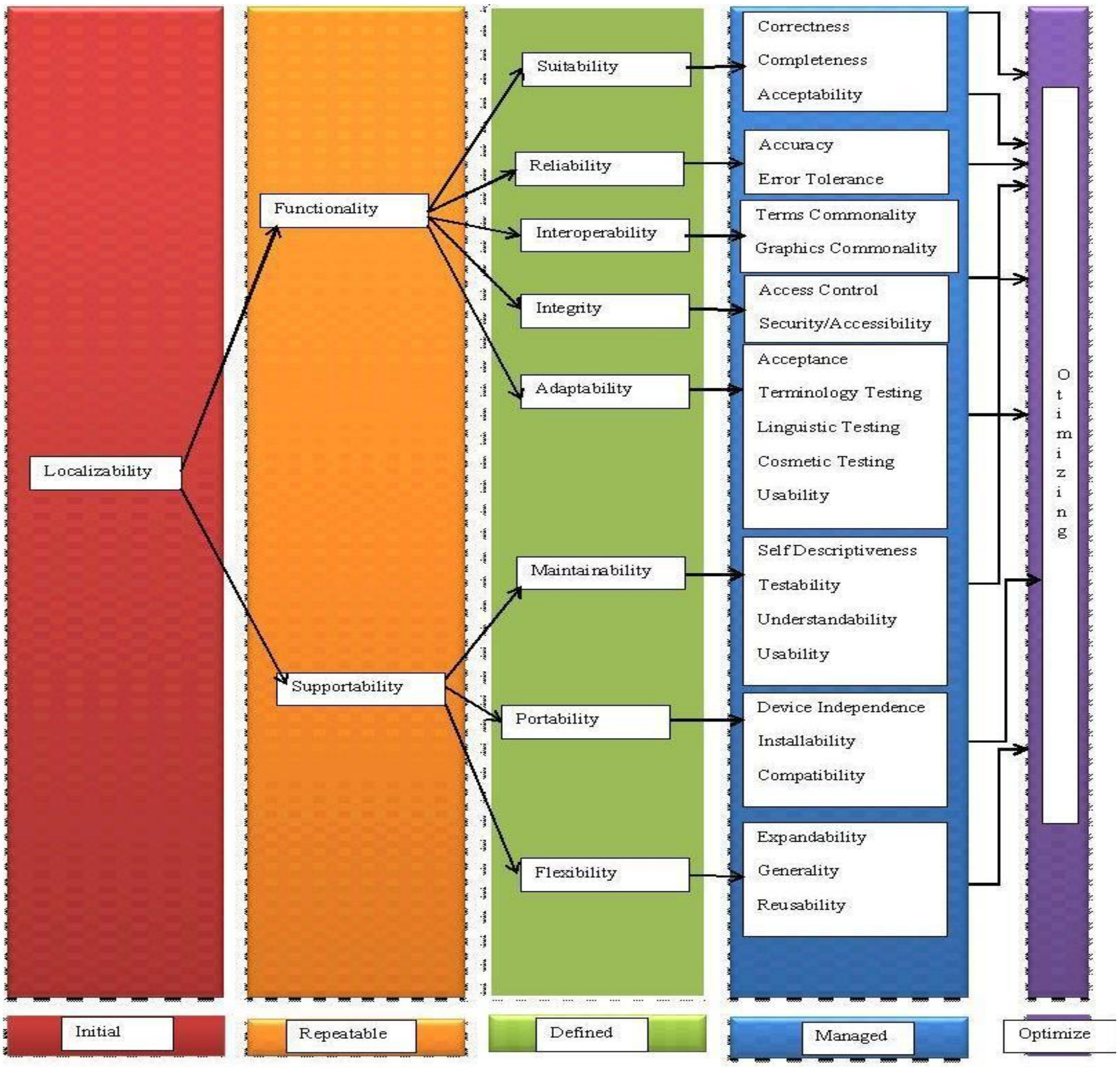


Figure 5: Generalized quality model adopted for localized product with advantages of Mc Call, Boehm, CMM, Furps, and ISO 9126

There are also quality model builder tool readymade available in the market that can create your own quality model like for creating a quality model in the form of a **tree** we need to provide the tool with information as its Name, priority, Relative ordering etc. and then we can have the result as Relative importance, Maximum value of node, Normalized result Rating level

1.3. Localizability: - The ability of a computer software program that can be converted to any specific or general local language according to its local culture with local look n feel.

1.3.1. Functionality:-The expected functions and features for which the local product is designed are either fulfilled or not.

a. Suitability :- The extent to which a local product is suitable for the particular local market

[1] Correctness: - The extent to which a local software meets its specification.

[2] Completeness:-The extent to which a local software product is Complete with regard to its functionality and its look n feel.

[3] Acceptability (For optimizing it, needs to be maximize):-The extent to which it is expected to be accepted by its local customers.

b. Reliability:-The extent to which the local software performs its intended function without failure.

[1] Accuracy(For optimizing it , needs to be maximize):- The extent to which information provided in the local system is accurate and reliable

[2] Error Tolerance:-The amount of error that can be tolerated by the system.

c. Interoperability:-The effort required to couple a localized system with other localized system.

[1] Terms Commonality:-Integration of two localized system based on the extent to which terms are common in both the system.

[2] Graphics Commonality:-Integration based on the common features in look n feel of both products according to the local market.

d. Integrity:-The protection of the software program from unauthorized access.

[1] Access Control:-Controlling the access by specifying access privilege to the customers of the local product with login passwords or digital signatures etc.

[2] Security/Accessibility: - Securing the software in case of power failure and authentication of its customer.

e. Adaptability:-The extent to which local product is adaptable to the local environment, people, technology and platforms.

[1] Acceptance:-The extent to which the local product is expected to be accepted by the local customer.

[2] Terminology Testing: - The extent to which the terms used in the local product matches with its regional locale.

[3] Linguistic Testing:-Checking for the context in which the specific term used and its language suitability.

[4] Cosmetic Testing:-Testing the look n feel of the local product in accordance to the local market and local customers.

[5] Usability:-The extent of the effort required to learn operate and understands the functions of the local software.

1.3.2. Supportability:-The extent to which the software product supports various features like its maintainability, portability, flexibility etc.

a. Maintainability:-The effort required locating and fixing errors in the software as a part of maintaining the system.

[1] Self-Descriptiveness:-The extent to which system specifies itself with help or easy to use menus and terms, so that it can be learned and used by the local customer easily.

[2] Testability:-The effort required to test the localized system to ensure that it performs its intended function without bugs n errors.

[3] Understandability:-The effort required to learn the local system so that its purpose is clear to its customer.

[4] Usability:-The effort required to learn, operate and to understand the local system by its users.

b. Portability:-The extent to which a local system can be transferred from one platform to other.

[1] Device Independence:-The extent to which a local system is independent to its device on which it is running.

[2] Installability:-The easiness of the local software product to be installed on different platforms with differing technologies.

[3] Compatibility:-The ease of using the local product with different computer system architectures.

c. Flexibility:-Modifiability of the local software to the extent that it facilitates the incorporation of changes.
[1] Expandability:-The effort required expanding the features and functions of the local system.
[2] Generality:-The extent to which the code and features of the specific local system can be used for general localized system.
[3] Reusability:-The extent to which the code can be reused in some different local system.

To test the quality model on a tester module we have a quality tester module named **Gunvatta Parikshak** that takes as input an application and tests it based on various criteria's like terminology testing, acceptance testing, cosmetic testing, effectiveness testing etc.

Note: **-** We will use GP as an acronym for **Gunvatta Parikshak**

Table 4
Description of Product operations with respect to **Gunvatta Parikshak** Localisation tool

Basis	***Description when applied on Gunvatta Parikshak***	***Values Obtained****
Correctness	Whether the application GP tests the input application's terms ,its look n feel according to the local market or not . It is correct if it checks all the criteria which are specified and for which it is designed.	Values lies between 3 and 4
Completeness	GP tests the application based on various basis and results in many artifacts showing the completeness of the application.	Values lies between 3 and 4
Acceptability	GP checks at what extent the input application is acceptable by its local customer.	Values lies between 4 and 5
Accuracy	GP checks at what extent the information provided in the input local application is according to its locale.	Values lies between 4 and 5
Error Tolerance	GP checks how much errors and bugs the input application can tolerate when adapted in a new local environment.	Values lies between 3 and 4
Terms Commonality	When there is a need to interoperate two different local applications then GP checks that how many terms in total are common in both the local application to make it easy to integrate.	Values lies between 3 and 4
Graphics Commonality	When there is a need to interoperate two different local application then GP checks that how much similarities are there in graphics of both the application	Values lies between 3 and 4
Access Control	GP checks for privileges that are being provided to the users or customers of the local application.	Values lies between 3 and 4
Security	GP checks for the level of security for the local application , security in the case of power failure, authentication etc.	Values lies between 3 and 4
Terminology	GP checks the terms that is being used in the local product are how much correct with regard to its original version.	Values lies between 4 and 5
Linguistic	GP checks for the context of the language used in the local product, whether the terms are right according to the context in which they are being used.	Values lies between 3 and 4
Cosmetic	GP checks for the cosmetic i.e. look n feel of the local product. At what extent it is according to the local culture and people.	Values lies between 4 and 5
Usability	GP checks for the effort required to learn the input local application whether it is expected to take an hour, a day or may be more or training required or not.	Values lies between 4 and 5

Self-Descriptiveness	At what extent the local software descript itself in form of help, tutorials , tooltips etc.	Values lies between 3 and 4
Testability	GP checks how efficiently the system can be tested like in how much time without having any bugs or errors.	Values lies between 2 and 3
Understandability	GP identifies the percentage value at which the local system is understandable by the local customer.	Values lies between 4 and 5
Device Independence	GP checks the local product whether it is independent of the device on which it is installed or not, and if yes then at what extent it is.	Values lies between 1 and 2
Installability	GP checks how easily the local software can be installed on the different platform like how much space, effort and time it requires to install.	Values lies between 3 and 4
Compatibility	GP checks for how much the input local application is compatible with different computer system architecture like the processor, RAM, OS, Memory that is being required.	Values lies between 1 and 2
Expandability	GP identifies how easily the local product can be expanded in a new version, or it may be how easily the features and functionality of the local product will be enhanced.	Values lies between 4 and 5
Generality	GP identifies the percentage value for the local product for being its general (i.e. being adapted for other local product) and for being its specific for the current locale.	Values lies between 3 and 4
Reusability	GP identifies the amount of the code that can be reused (like 10% or 20% etc.) in designing or maintaining some other or same local application.	Values lies between 2 and 3

*These values obtained are with respect to **Gunvatta parikshak** localization tool.
Here number represents 1- very low, 2- low, 3 - neither low nor high, 4 - high, 5 – very high.

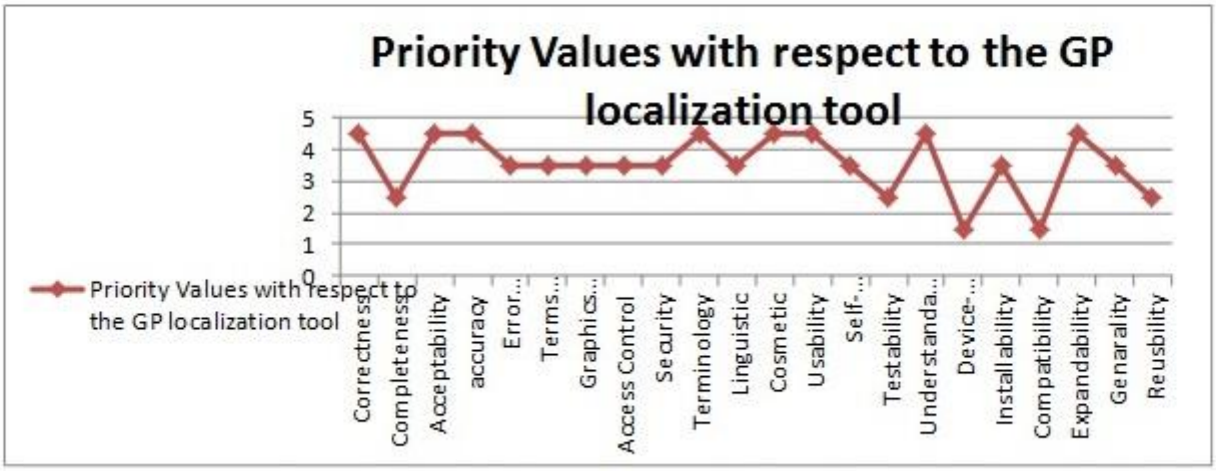


Figure 6: Priority values for Product Operations with regard to Gunvatta Parikshak

V. CONCLUSION

The above paper presents a generalized quality model in both a summarized and detailed view that combines the advantages of various existing quality model adopted for localized product, as it considers quantitative factor of Mc-Call quality model and qualitative factor of Boehm software quality model. From "Fig 6 " we can conclude that for 'Generalized quality model for localized product' higher priority factors are its Correctness, Acceptability, Accuracy, Terminology, Cosmetic, Usability, Understandability and its

Expandability and the other factors like portability, Install ability, Integrity are with lower priorities i.e. they are required but acceptable with minor changes.

The paper presents various quality factors of product operations with the help of a quality tester module name **Gunvatta Parikshak** that specifies them together with the corresponding value metric that ranges from very low to very high.

REFERENCES

[1] K.K. Aggarwal & Yogesh Singh ,' Software Engineering ' (3rd ed., 2007, New Age International Publishers)

[2] Roger S. Pressman, 'Software Engineering ' (5th ed. 2000 McGraw-Hill Higher Education Publishers)

[3] 'Software Quality models and philosophies', '[Online]' http://www.bth.se/com/besq.nsf/(WebFiles)/CF1C3230DB425EDCC125706900317C44/$FILE/chapter_1.pdf(Accessed 19thSep 2013)

[4] 'LISA QA Model 3.1 -- Assisting the localization development, production and quality control processes for global product distribution','[Online]', http://dssresources.com/news/1558.php

[5] 'Localisation Quality Assurance-How Important?','[Online]', http://www.globalvis.com/localization-qa- Dan Johnson "The Guide for multilingual computing and technology LOCALISATION " July-Aug 2003 , 18-20,'[Online]' http://www.multilingual.com/downloads/screenSupp57.pdfhow-important/