CHAPTER 5

ENHANCED SOFTWARE REUSABILITY METRICS FOR USER CENTERED DESIGN

5.1 INTRODUCTION

The software reusability has been considered as a quality metric that explores the extent to which the software product can be reused by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. The software reusability engineering provides features to determine the user ability in a specified environment within a usage time to bring maximum Return on Investment (ROI) for the software organization. It also provides a Software Process Improvement (SPI) for the product. The output of such an engineering exercise is to identify and inform the developers of the organization about the accessibility of the product for maximum reuse within minimal time.

The software reusability is considered as a combined form of design for reusability and reusable design with understandability, learnability and operability of the product to the final user or user group (Pressman 2001). Context aware computing like user identity based services and location aware appliances like GPS based services are to be considered in determining the software reusability (Carroll 2002).
The ubiquitous computing advocates that the Human-Computer-Interface (HCI) brings the information in the style like 'wwiwwa' (we want information where we are) instead of 'Wysiwyg' (what you say is what you get). New developments in the fields of multimedia, hypertext and virtual reality technology face major challenges in designing HCI which in turn affect the reusability of these products. Roomware is a concept which necessitates a different approach for determining the reusability of a component in a device oriented conceptual cooperative computing environment.

User performance measurement is carried out using a number of sub measures and these are: task effectiveness, efficiency, productive period and relative user efficiency (Chang 2006). Therefore software reusability measures and the design for reusability are centered on the users, necessitating A User Centered Design (UCD) model that can be arrived by analyzing the important user entities and their corresponding attributes mapped to the design attributes. But the UCD model may include the information interface on which a user is utilizing the power of a software product and the context of usage for timely behavior of the product. Hence an enhanced software reusability model is proposed in this research work by considering all the modern computing devices and their environment styles with attractiveness and of advising type.

The main objective of this work is to propose an enhanced software reusability model and develop a graphical tool for calculating reusability attributes for User Centered Design (UCD) applications. The software reusability analysis is performed based on the various aspects of Return On Investment (ROI) and an acceptable set of quality attributes of modern software quality assurance activities lead to products. The design model is mapped to an enhanced design model to achieve the specified goals in a
specified context of reuse. The software reusability attributes are so engineered with other quality attributes to improve the quality-in-use of the product within the scope of usage time and user abilities. Various software reusability metrics are also proposed to relate the user capability, enterprise benefits, information interface and user application along with the corresponding physical spatial architecture. The enhancement of software reusability is so modeled as to support the required devices utilized in the human computer interactions.

5.2 SOFTWARE REUSABILITY ENGINEERING

The software reusability engineering may be considered as a discipline which concerns with the analysis of non-functional characteristics of the software product influenced by the user group within the usage period keeping the operating environment and the functional requirements of the application unchanged. Such engineering leads to determine the responsiveness and the accessibility factors of the product. The conventional design model considers the user application focusing its requirements and user ability to handle the software product up to satisfaction in usage time. The various user-centered design factors are listed in Table 5.1. To arrive at a reusability design model, the design factors are to be weighted and suitably mapped into a set of enhanced reusability design factors like ROI, (Return on Investment), SQA (Software Quality Assurance), P-CMM (Peoples Capability Maturity Model) are shown in Table 5.2.
Table 5.1 Reusability Design Factors

<table>
<thead>
<tr>
<th>S.No</th>
<th>Design attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reuse case</td>
<td>Customer requirements - interaction between application and agency</td>
</tr>
<tr>
<td>2</td>
<td>Reuse group</td>
<td>Types of Organizations - small, medium, large size</td>
</tr>
<tr>
<td>3</td>
<td>Reuse purpose</td>
<td>Purpose of computation - accessibility, responsiveness, comprehension, correctness</td>
</tr>
<tr>
<td>4</td>
<td>Reuse time</td>
<td>Frequency of access - occasional, regular, often, always</td>
</tr>
</tbody>
</table>

Table 5.2 Enhanced Reusability Design Factors

<table>
<thead>
<tr>
<th>S.No</th>
<th>Design attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reuse Interface</td>
<td>Visual requirements, control, presentation and interaction between applications</td>
</tr>
<tr>
<td>2</td>
<td>Reuse context</td>
<td>Truly ubiquitous - time, behavior, functional and architecture</td>
</tr>
<tr>
<td>3</td>
<td>Reuse work Space</td>
<td>Dimension of Investment, Quality assurance and Process improvements</td>
</tr>
<tr>
<td>4</td>
<td>Reusing people and reuse devices</td>
<td>Nature of computing device - interactive, adaptive and predictive and Collaborative people</td>
</tr>
</tbody>
</table>

The mapping from conventional UCD model for reusability to enhanced UCD model of reusability is based on the reuse environment in which the software product is put into use. The design is focusing on the reuse environment in which the software product is used which is not clearly defined
in the earlier works and suitably incorporated in the enhanced reusability design model.

5.3 ENHANCED REUSABILITY

One of the major contributions of thesis is the newly proposed enhanced design model in which not only the impact of context of reuse and nature of ubiquitous computing but also the nature of co-operative devices and the collaborative people are considered. The multi users belonging to different geographical areas who are involved in applications that share the information in an asynchronous manner through flexible user interfaces may also taken into consideration. Moreover, a graphical tool has been developed in Visual Basic to determine the minimum and maximum values of the enhanced goal metrics for different reusability matrices. The reuse group and reuse time are mapped into the design model into user interface and device attributes respectively. The software reusability design has been enhanced by incorporating the above mentioned factors in a Matrix Model (MAM) of software reusability, which is another major contribution of this research work.

Business aspects to achieve the user goal which comprises of ROI of company, user satisfaction, product quality assurance in the perspectives like safety and the product's effectiveness, productivity are considered. This mathematical approach leads to provide an overall picture of the relatedness of user and product. The software product reusability matrix has been constructed by considering the core factors of the product reusability in terms of four new metrics that were proposed in this research work namely Attribute Reuse Factor (ARF), Method Reuse Factor (MRF) and Attribute Reuse Factor with Inheritance (ARFI), Method Reuse Factor with Inheritance (MRFI) which is as [R] and this is a 1 x 4 matrix. The other matrix represents the
Design Matrix [D] that considers the factors in various dimensions to make a 4x4 matrix. The multiplied effect of the two matrices helps the user to achieve the socio-business goals with ease and satisfaction. The resultant [G] is a 1 x 4 matrix whose elements reflect the end user goal in handling any software product after reuse techniques.

5.4 REDESIGN FLOW AND ITS ATTRIBUTES

The software product is highly reliable and maintainable if the design attributes in all dimensions are selected and categorized to operate on the reusability. The functional design attributes like the number of processes and input/output parameters along with the amount of external data are combined with ARF and MRF in the application considered. The behavioral attributes like the number of states of individual objects and the number of transitions are combined with ARFI and MRFI. The architectural design parameters and the operational environment parameters are also considered for the overall design matrix to determine the impact of reusability at the design time itself. The modified design values determine the cost associated with the development and also maintenance. This type of development technique prior to coding stage helps to complete the development of the software product of the given application within the stipulated time. The maintenance activities either in the form of corrective or enhancement type accounts for the enhanced usability of the product at approximately zero risk. At the same time, the working environment will bring safety and satisfaction to the end user in the case of multiple user application areas like E-Commerce and Distributed Banking applications. The reusability factors are to be centered in the case of enterprise application development since the repetitiveness of the same type of work in a lonely room even with high end computing resources will bring fatigue and mental tiredness easily.
5.5 ENHANCED PRODUCTIVITY

The modern software reusability is depending upon the software environments such as full functional component libraries and sharable information stored in the distributed systems in corporate buildings with all sorts of informational resources. Interactive environment components like reusable interfaces improve the information sharing between software team members and users in which they share the large information base which helps in creating and deleting objects in a mode-less interaction. This sort of enhancement through the information sharing supports remote interaction with other reusable components. The software productivity is totally depending on the efficient methods in reusing the modules within the available time especially in the case of the text application software or the graphics application software. Efficiency is responsible for its productivity measure in these scenarios where the reusability factor is itself depending on the techniques used in utilizing the available hardware and software resources efficiently. The usage time and various resources that are to be shared in some of the application areas are also responsible for enhancement of the existing reusability model thereby the overall productivity is improved by that software product. The reuse model is proposed as Reuse-Design-Goal Matrix model in determining the effect of software reusability is shown in Figure 5.1.
Figure 5.1 Reuse - Design - Goal Model

Table 5.3 Enhanced Reusability attributes and other Design Factors

<table>
<thead>
<tr>
<th>Design Factors D</th>
<th>Reusability attributes R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Design Parameters</td>
<td>Processes</td>
</tr>
<tr>
<td>Behavior Design Parameters</td>
<td>States</td>
</tr>
<tr>
<td>Architecture Design Parameters</td>
<td>Components</td>
</tr>
<tr>
<td>Environment Design Parameters</td>
<td>Networks</td>
</tr>
</tbody>
</table>

The reusability matrix [R] consists of ARF, MRF, ARFI and MRFI of the product. The design matrix [D] consists of four columns representing
functional, behavioral, architectural and environmental design parameters respectively. The row elements of matrix \([R]\) when interacted with the individual column elements of matrix \([D]\) yields the resultant matrix \([I]\) which leads to satisfaction, effectiveness, productivity and easiness elements of the product when the end user handles the product in specified environment.

The above reusability model may be represented mathematically as follows

Mathematically,

\[
[R] \times [D] = [I]
\]  

(5.1)

The enhanced reusability model turns out to be a resultant matrix of the modified matrix \([R]\) into \([D]\) where the elements of original \([R]\) has been transformed as per the reuse information model, reuse architectural space, reuse behavioral context and user interactive device or people. The enhanced Reuse-Context-Goal Matrix model of software usability can be given by the equation,

\[
[R] \times [D] = [G]
\]  

(5.2)

where the enhanced reusability design attributes are considered in the matrix \([R]\) and the resultant goal matrix as shown in Table 5.3.

The proposed software reusability metrics are useful in the emerging technologies as well as in new application areas demanding new approaches like multi user applications supporting cooperative software development. Such enhanced reusability metrics are listed below:
1. **Reusability @ Return On Investment** which indicates the extent at which the software product is effective in terms of returning the investments for the user towards his accurate, secured functionality on multiple computing platforms in suitable applications. This is due to thorough understanding of the product in achieving the goal of computation.

2. **Reusability @ Productivity** which brings the productivity of the software within the operational time through an expected behavior style utilizing the minimal resources. This can be very well enhanced if the operability of the same is relatively high.

3. **Reusability @ Satisfaction** which indicates the extent at which the software can give end user satisfaction in a reliable, fault tolerant and recoverable manner with perfect maturity. This is possible only if the software is learnable irrespective of the user group.

4. **Reusability @ Maintainability** is the metric of the software product in its usability domain through which safety aspects in completion of the work may be determined. In turn, this approach necessitates the analyzability and testability of the product when there are essential major or minor changes that have to be carried over for a stable functionality. This metric speaks the extent at which the end user feels easy in handling and navigating through the software with or without the 'help' documents and finds the adaptable nature of the product in replacing from one system to other system and installing with other coexisting software including dual operating systems, plug-ins and open source software products.

5. **Reusability @ Complexity** is the metric of software complexity that is computed by considering the ratio of the percentage error with reuse and percentage. This implies that complexity can be assessed with error rate. From this it is observed that the error rate increases with the increase in complexity.
A Reusability calculator has been developed in Visual Basic which will be helpful in calculating the end user reusability values after each incremental value that is selected from a survey conducted among different groups of end users. The enhanced reusability design attributes and the user goal attributes are shown in Figures 5.3 and 5.4. A single unit of software reusability measurement can be found out as the product of all the elements of the goal matrix as for a fixed period of use time by a specified user group.

\[ \text{Net Reusability} = \text{ROI} \times \text{Satisfaction} \times \text{Productivity} \times \text{Easiness} \times \text{complexity} \]

(5.3)

As a Sample case

\[ R \times D = G \]

Where \( R, D, C \) are,

\[ R = [ \text{ARF, MRF, ARFI, MRFI} ] \]

\[ D = \begin{bmatrix} \text{Processes} & \text{Data Items} & \text{Entities} & \text{Stored} & \text{Data} \\ \text{States} & \text{Transitions} & \text{Guards} & \text{Nests} \\ \text{Components} & \text{Connectors} & \text{Interfaces} & \text{Layers} \\ \text{Networks} & \text{OSystems} & \text{Con. Users} & \text{Links} \end{bmatrix} \]

\[ G = [\text{return on investment, satisfaction, productivity, maintainability}] \] By giving random values to all the attributes,

\[ R = [0.2 \ 0.2 \ 0.4 \ 0.2] \text{ Calculated from the table A.I.3 & 5} \]
As per equation (5.3), that is \( G = R \ast D \)

\[
D = \begin{bmatrix}
0.2 & 0.2 & 0.4 & 0.6 & 0.3 \\
0.4 & 0.6 & 0.1 & 0.1 & 0.2 \\
0.2 & 0.1 & 0.4 & 0.1 & 0.2 \\
0.2 & 0.1 & 0.1 & 0.2 & 0.3 \\
\end{bmatrix}
\]

\[ G = 0.2400 \times 0.2200 \times 0.2800 \times 0.2200 \times 0.2400 \]

The single metric for reuse driven end user goal is the product of all the elements in G matrix that is 0.00078

By incrementing ARF value by 0.1,

\[
\begin{align*}
R^* \ast D &= G \\
R^* &= \begin{bmatrix} 0.3 & 0.2 & 0.4 & 0.2 \end{bmatrix} \\
D &= \begin{bmatrix}
0.2 & 0.2 & 0.4 & 0.6 & 0.3 \\
0.4 & 0.6 & 0.1 & 0.1 & 0.2 \\
0.2 & 0.1 & 0.4 & 0.1 & 0.2 \\
0.2 & 0.1 & 0.1 & 0.2 & 0.3 \\
\end{bmatrix}
\end{align*}
\]

As per equation (5.4), that is \( G^* = R^* \ast D \)

\[ G^* = 0.2600 \times 0.2400 \times 0.3200 \times 0.2800 \times 0.2700 \]

The enhanced goal attributes are given in G matrix. The single enhanced reuse driven goal metric is the product of all the elements in G matrix that is 0.001509. From this we observe that whenever there is a change by 0.1 in MRF the corresponding change leads to enhancement of 193% in the user goal. As per the survey results the end user goals are
enhanced if the ARF of the software due to the factors depending on processes and components with their interfaces. The Relationship between enhanced reusability factors and the modified goal factors are shown in Figures 5.2 to 5.5.

The enhanced reuse metrics are proposed to represent the satisfaction of the specified set of customers not only in terms of their financial requirements or business satisfaction but also in product quality terms through repair and maintenance. In the equations 5.5 to 5.9, the relationships are shown.

\[
\text{Reuse\textregistered} \text{ ROI} = \frac{\text{Difference in Cost } Dj}{\text{Development Cost without Reuse } Ci} \quad (5.5)
\]

\[
\text{Reuse } \oplus \text{ Productivity} = \frac{\text{Size Difference } Sij}{\text{Size without Reuse } Cij} \times \text{Time} \quad (5.6)
\]

\[
\text{Reuse } \oplus \text{ Satisfaction} = \frac{\text{Improvement in Usability } Uj}{\text{Usability without Reuse } Ci} \quad (5.7)
\]

\[
\text{Reuse } \oplus \text{ maintainability} = \frac{\text{Difference in Repair Cost } Dt}{\text{Repair Cost without Reuse } Ci} \quad (5.8)
\]

\[
\text{Reuse } \oplus \text{ complexity} = \frac{\text{Percentage Error with Reuse}}{\text{Percentage Size difference with Reuse}} \quad (5.9)
\]

The five reuse driven business software metrics are useful in determining the outcome of any software organization focusing on reuse of software processes. The metrics are to be refined for different set of developers and testing personal. Based on the above calculations, the software
process and people improvement activities may be carried over to raise maturity level of over all members in any organization.

Figure 5.2  Impact of Reuse on ROI, Productivity, Satisfaction

Figure 5.3 Comparison of Satisfaction, Maintainability, ROI
Figure 5.4 Comparison: ROI, Satisfaction, Complexity

Figure 5.5 Comparison: ROI, Productivity, satisfaction, Maintainability
5.6 CONCLUSION

The reusability evaluation has been carried out and a new matrix model of enhanced reusability attributes has been achieved. The reusability attributes are given specific increments from the survey conducted and the relationship between various reuse driven goals attributes and enhanced reusability attributes like ARF and MRF are shown in various graphs. A graphical simulator has been developed to find the impact of reuse on the end user goal attributes. The single unit of reuse driven software goal attribute is also proposed incorporating all the factors regarding the environment, the devices of computation and the information interface as per the context of usage of the product.