Role of Formalism in Software Reusability's Effectiveness

Muhammad Ilyas and Mubashir Abbas

*Department of CS & IT, University of Sargodha,
Sargodha Pakistan
m.ilyas@uos.edu.pk, mubashir.abbas@yahoo.com*

Abstract

Economical, qualitative and shorten time software development are key objectives in software engineering. Different techniques are used to achieve entire goals. Reusability is one of the popular software development methodologies, which effectively reduces time, cost and effort for software development. It also minimizes software failure risk by using already tested components. The objective of minimal time, cost and effort can be fulfilled and maximized by effective use of reusability. This effectiveness can be achieved by formalizing each activity during reusability process. Adaptation of formalism relatively enhances effectiveness of reusability methodology. In this paper we have investigated some factors about reusability effectiveness and role of formalism in its effectiveness. In this investigation 42 factors are taken into consideration. These factors are grouped into 10 sections like Reusability Process, Reusable Test, Formalism and Extraction of Reusable components etc. Our findings are based on statistical analysis of industrial data that indicates the way in which reusability is taking place and productivity is earned.

Keywords: *Reusability Process, Reusability Methodology, Formalism, Extraction, Effectiveness*

1. Introduction

Reusability has moved towards an economical software development paradigm. Due to its significant paybacks it has become an ideal approach for software development. Its effective uses can double the paybacks in terms of less time, cost and effort. Along with improvement in productivity it also minimizes risk associated with software success [4]. This effectiveness can be achieved and improved by adopting formalism during reusability process. To implement such formalism quality criteria should be defined for each activity during reusability process. For this purpose reusability process is divided into three stages:

- Extraction of Reusable Components
- Storage of Reusable Components
- Deployment of Reusable Components

This study is restricted to Extraction stage only. Storage and Deployment stages of reusable components are not under discussion here. Formalism is applied at Extraction level only and results are gathered. For this purpose a Reusable Test is conducted at Extraction stage. This Reusable Test consists of certain metrics to earn healthy reusable components for repository. It will help to maximize productivity and efficiency of overall system.

Reusable Test at Extraction stage consists of following metrics:

1. Versatility Metric
2. Reliability Metric

3. Understandability Metric

1.1. Versatility Metric

Versatility metric is defined as the ability to perform multiple tasks simultaneously by a software component. It is versatile in its functionality. It can be used in various environments with different requirements. Software component with versatility feature contains higher value of flexibility. Here Versatility metric is used as a combination of Generality and Portability Metrics.

- Generality Metric
- Portability Metric

1.1.1. Generality Metric: This metric is used to measure general behavior of software components. W. J. Salamon [5] evaluated software component on the basis of following metrics:

- Multi Usage
- Data Volume Component
- Data Valued Component
- Mixed Function

K.K Aggarwal [6] measured component's generality by using following metrics:

- Function Template
- Class Template

Presence of above-mentioned characteristics in a software component supports its general behavior.

1.1.2. Portability Metric: This metric is used to analyze that component is capable to work on different platforms in terms of hardware and software. Components are evaluated that how easily they can be moved from one environment to another. W. J. Salamon [5] presented following metrics to check component's portability characteristics

- No. of compatible systems
- No. of system utilities utilized
- No. of modules making I/O reference
- No. of modules not following conventions , programming languages and tools are available

1.2. Reliability Metric

This metric is used to investigate the component's performance history in different circumstances for the given time period. It evaluates that component has performed required functionality satisfactorily. Analysis is performed by recording

- Failure History
- Error Rate
- Error Type
- MTTF (Mean Time To Failure)

1.3. Understandability Metric

Readability metric is used to examine understandability, readability and clarity of a software component. Bajeh [7] used following metric to measure readability of given software

- No. of lines of code properly indented

In addition to the above metric comments metric can also be useful to measure readability of a software component.

- No. of lines of code properly commented

Moreover, along with the proper extraction of reusable components, implementation of formalism is required at classification and deployment stages during reusability process. Aim is to formalize the each activity during reusability process. It will help to maximize paybacks and to minimize overall effort.

The focus of this paper is to measure the effectiveness of reusability methodology and role of formalism in its effectiveness. For this purpose we have analyzed existing usage level of reusability and role of formalism in its effectiveness in different software organizations. A questionnaire was designed and survey was conducted to gather relevant data about reusability methodology. This survey makes analysis of:

1. Reusability methodology effectiveness.
2. Benefits earned by adopting reusability.
3. Existing Reusability Process.
4. Role of formalism in Reusability Process.
5. Extraction of Reusable Components

Rest of the paper consists of the following sections: In Section 3, we have proposed a conceptual model that describes different stages during reusability process. In Section 4, Hypothesis is given, In Section 5 data collection information is given and in section 6 we have discussed the results based on our findings and the final section concludes our work directions for future perspective.

2. Prior Literature

Research on software reusability has been in progress since long time, Researchers have worked almost on every aspect of this methodology. Early studies have presented reusability analysis, metrics, models, advantages and different tools to use it effectively. This aspect, effectiveness of reusability paradigm is the core of new study in this field.

Arun [1] found that reusability significantly contributes in lessening time, cost and effort for software development. They said that reusable component is developed once but it can save efforts for subsequent developments.

Farhan [2] discovered that although reusability reduces time, cost and effort but its implementation techniques are not controlled and managed. They suggested a framework to formalize this reusability process, so good compatible components can be recorded for reuse purposes.

Ramachandran [3] said that software reuse is the key to gain in productivity but problem is to identify and record quality reusable components. He developed a prototype known as Reuse Assessor and Improver System (RAIS). It can identify, analyze, asses and modify abstraction plus attributes and architectures to support reuse.

3. Research Methodology

This research is conducted to investigate the effect of reusability and role of formalism in its effectiveness. Here, a proposed conceptual model has been presented for this survey, then to analyze outcomes of this survey; hypothesis has been given followed by data collection and results discussion.

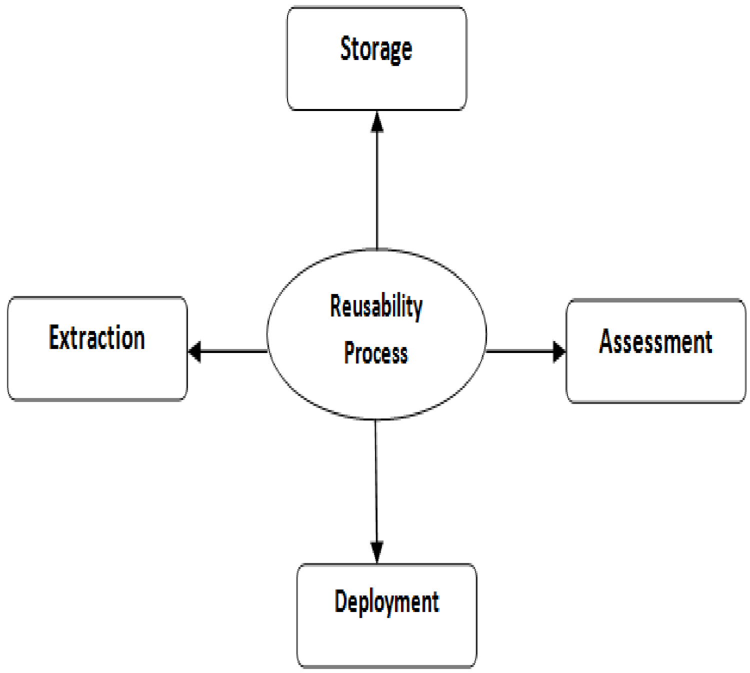

Figure 1. Reusability Process

3.1. Conceptual Model

The above figure presents a conceptual model which describes stages during reusability process. For better understanding reusability process is divided into three main stages: extraction, storage and deployment of reusable components. One intermediary stage assessment stage is used between storage and deployment stage. Objective is to thoroughly study the whole process to improve overall throughput. Further explanation of above stages is given below:

3.1.1. Extraction Stage: The process of earning quality software components for reusable repository is called Extraction Process. Identification and extraction of quality component is umbrella task of reusability management [8]. Idea is to earn healthy components for reuse purpose. To make extraction process efficient a reusable test is conducted at this stage. This reusable test consists of following significant metrics:

- Versatility Metric
- Reliability Metric
- Understandability Metric

3.1.2. Storage Stage: It is the process of properly classifying and storing reusable components in the reuse bank. It will help in express availability of suitable software components for the given requirement. Availability of required component in short time will increase overall efficiency.

3.1.3. Assessment Process: It is the process to find optimal software component among given availability. It selects the component on the bases of following criteria:

- Maximum Similarities
- Minimum Modifications

As the component with maximum similarities and minimum modification is highly suitable in terms of compatibility and integration cost.

3.1.4. Deployment Stage: It is the final stage which integrates selected reusable component with new system. This stage evaluates the component on the bases of

- Compatibility
- Minimum integration cost
- Level of confidence/Trust

On the availability of above characteristics components is finally integrated to the new system.

3.2. Hypothesis

We have developed a hypothesis to analyze the impact of reusability that has a significant effect in software development.

The dependent variable is:

Y= Reusability Process

Here $Y=Y_1+Y_2+Y_3$……………………………………..Y_{12}

Y_1=Reusability is a productive approach for software development

Y_2=Reusability is an economical software development methodology

Y_3=Reusability is just a practice of rework

Y_4=Reusability is not a creative methodology

Y_5=Reusability reduces time required for software development

Y_6=Reusability reduces cost required for software development

Y_7=Reusability reduces effort required for software development

Y_8=Reusability reduces risk associated with software success.

Y_9=Reusability process in software organizations is formal

Y_{10}=Reusability process in software organizations is satisfactory

Y_{11}=Reusability process in software organizations is conventional

Y_{12}=Reusability process in software organizations is requiring improvement

The independent variables are:

X_1=Formalism is quite necessary during reusability process

X_2=Formalism can double the reusability benefits

X_3=Formalism can put a very little improvement

X_4=Formalism will increase time limits for development

X_5=Formalism consequences are less than its effort

X_6=Extraction process earns quality software components for reusable repository

X_7=Extraction process is unnecessary, its usage will make reusability process complex

X_8=Reusable Test is implemented for Extraction of reusable components
X_9=Reusable test checks software component's reliability
X_{10}=Reusable test checks software component's portability
X_{11}=Reusable test checks software component's generality
X_{12}=Reusable test checks software component's understandability
X_{13}=Multi Usage software components are considered to be more general
X_{14}=Multi Function software components are considered to be more general
X_{15}=Data Valued software components are considered to be more general
X_{16}=Data Volume software components are considered to be more general
X_{17}=Components containing Function Templates, are considered to be more general
X_{18}=Components containing Class Templates, are considered to be more general
X_{19}=Portability can be measured by finding total no. of compatible systems
X_{20}=Portability can be measured by finding total no. of system utilities utilized by software component
X_{21}=Portability can be measured by finding availability of common and subset of languages used
X_{22}=Portability can be measured by finding availability of programming languages and tools used
X_{23}=Portability can be measured by calculating total no. of modules making I/O reference
X_{24}=Portability can be measured by calculating total no. of modules not following conventions
X_{25}=Proper Indentation of source code lines enhances readability
X_{26}=Proper comments on source code lines enhance readability
X_{27}=Reliability can be determined by Errors/Failures History
X_{28}=Reliability can be determined by recording Error types
X_{29}=Reliability can be determined by recording Error reasons
X_{30}=Reliability can be determined by recording Error rate
To study the impact of independent variables on reusability process, the following hypotheses have been developed.

3.2.1. Null Hypothesis (H0): None of the independent variable has a significant effect on the dependent variable, *i.e.*, all the regression coefficients are zero.

3.2.2. Alternate Hypothesis (H1): At least one of the independent variables has significant effect on the dependent variable i.e. at least one of the regression coefficients is not equal to zero.

To test the hypothesis we have applied Multiple Regression Technique using SPSS 17.0 statistical package. The Level of significance (α) for this study is =0.05.

4. Data Collection

For this study, a survey type questionnaire was designed, in which participants were asked about the usage of reusability methodology. Aim was to investigate reusability process in software development.

The questionnaire was categorized into 10 different groups. Each group contains questions that measure the usage of reusability, its implementation methodology and adaptation of formalism to maximize its benefits. Overall aim was to investigate impact of reusability in software development.

Table 1. Summary of Data Collected from Different Organizations

Data collected from organizations under study	
Total no. of organizations	17
Average organization size	70 employees
Total no. of respondents	27
Respondent information	Project Manager Or Team Leader=11 Technical Member(SEPT) = 11 Software Engineering Process Group (SEPT) Member Others=5
Organization information	Software Developers=10 Telecommunication=1 Statistical Analysis center=1 Information Centers=2 Others=3

This questionnaire consisted of subjective question to measure reusability quantitatively. An ordinal scale was designed to gather qualitative data. The respondents were asked to use the numeric values for response. The scale used five choices [1-5], each having equal intervals [1=lowest and 5= highest].

For data collection, questionnaire was distributed among 50 different organizations. Out of these 50 organizations 27 people from 17 organizations responded. More details about collected data from different organizations understudy has been given in Table 1.

5. Results Discussion

This section presents results of Regression Analysis applied on the collected data by describing normality of data, the reliability of model, relationship among variables and the selected significant variables.

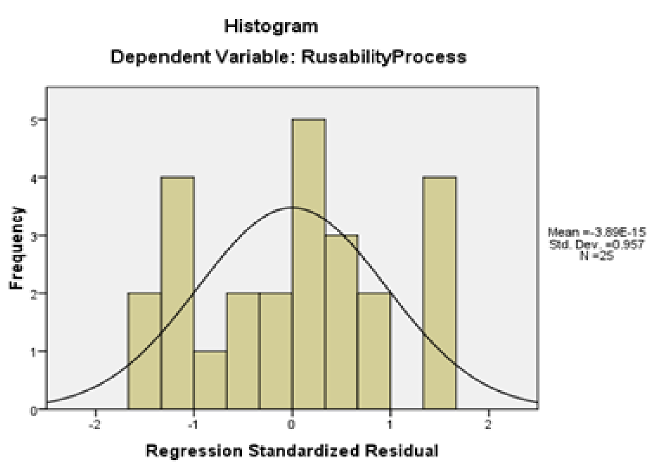

Figure 2. Observed Frequency Distribution of Dependent Variable w.r.t Normal Frequency Distribution Curve

5.1. Graphical Analysis of Normality

In Figure 2, we have put up a histogram of the collected data and compared it with normal curve. The observed histogram shows a normal trend; therefore we consider that data is normal. Although it is an approximate method but it produces a very clear picture of data.

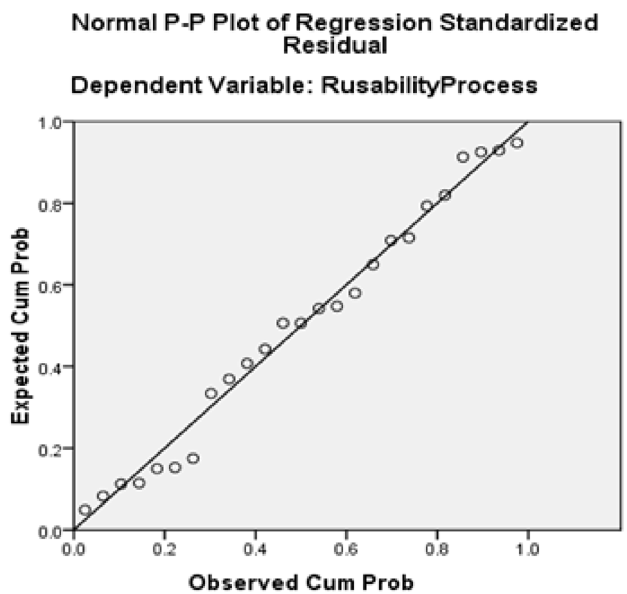

Figure 3. Normal Probability Plot of Dependent Variable using Observed and Expected Cumulative Probabilities

However, to ensure the normality of the data, we have constructed Normal Probability Plot. This is more reliable presentation as compared to histogram because it compares cumulative distribution of the actual data with the cumulative distribution of a normal distribution. In Figure 3 the straight diagonal line represents a normal distribution. The squares on the diagonal line represent actual data. The comparison between two lines shows that actual data line is almost straight and tends towards normality.

5.2. Testing Reliability of Model

Table 2 shows the results of multiple regression analysis. The entry labeled R is the correlation coefficient between the observed value of dependent variable and the predicted value based on regression model. 1 value of R tells that dependent variable can be predicted completely from the independent variable. It is the ideal case. In our analysis value of R is 0.693, so this regression model correctly predicts up to 70%. R Square known as Coefficient of Determination, is the variation in the dependent variable (Y) that can be explained by independent variables (X_1,X_2,X_3……………………X_{30}); provided all the other independent variables are kept constant. In Table-2, R Square is 0.481, so approximately 49% of the variation in reusability process is explained by the independent variables.

The observed significant value in Table 2 is less than α (where α=0.05), which describes the reliability of the model. Also significant value of F test is 0.024 which is less than 0.05 so the null hypothesis is rejected and we can say with 95% confidence that not all the independent variables are insignificant.

Table 2. Summary of Regression Model

				Change Statistics				
R	R Square	Adjusted R Square	Std. Error of the Estimate	R Square Change	F Change	df1	df2	Sig. F Change
.584[a]	.341	.312	.34555	.341	11.905	1	23	.002
.693[b]	.481	.433	.31367	.140	5.913	1	22	.024

a. Predictors: (Constant), Formalism
b. Predictors: (Constant), Formalism, Extraction
c. Dependent Variable: Reusability Process

5.3. Examining Relationship among Variables

The observed significant values of F test in Table 3 are less than α (where α=0.05). Therefore, it can be concluded that there is significant relationship between dependent and independent variables.

The independent variables that have a significant affect (Positive/Negative) on dependent variables are displayed in Table-4.

Independent Variable Formalism (X_1, X_2……..X_5) and Extraction (X_6, X_7………X_{12}) having significant values less than α (0.05) are selected and other insignificant variables are discarded.

Independent Variable Formalism (X_1, X_2……..X_5) and Extraction (X_6, X_7………X_{12}) Values in Table-4 determine that they have positive effect on dependent variable Reusability Process.

Table 3. Result of ANOVA applied on the Regression Model

Model		Sum of Squares	Df	Mean Square	F	Sig.
1	Regression	1.421	1	1.421	11.905	.002[a]
	Residual	2.746	23	.119		
	Total	4.168	24			
2	Regression	2.003	2	1.002	10.181	.001[b]
	Residual	2.164	22	.098		
	Total	4.168	24			

a. Predictors: (Constant), Formalism
b. Predictors: (Constant), Formalism, Extraction
c. Dependent Variable: Reusability Process

If we denote independent variable Formalism (X_1, X_2……..X_5) with X_F and Extraction (X_6, X_7………X_{12}) with X_E then our Regression Model produces following equation:

Effect of Reusability=1.457+ 0.275 X_F + 0.330 X_E ……………………….. (Equation-A)

The value of all regression coefficients (β) in the above equation is positive, which shows that both variables positively contribute towards the dependent variable; means they have positive effect on reusability process.

Table 4. Significant Variables Selected in the Regression Model

Model		Un-Standardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	T	Sig.
1	(Constant)	2.345	.355		6.608	.000
	Formalism	.363	.105	.584	3.450	.002
2	(Constant)	1.457	.487		2.991	.007
	Formalism	.275	.102	.444	2.702	.013
	Extraction	.330	.136	.399	2.432	.024

a. Dependent Variable: Reusability Process

By examining the regression model (Equation-A), 12 independent variables out of 30 have significant effect on dependent variable, all other 18 are insignificant. The significant factors are mentioned below:

Formalism (X_F- Regression Coefficient=0.275)

- Is quite necessary during reusability process
- Can double the reusability benefits
- Can put a very little improvement
- Will increase time limits for development
- Consequences are less than its effort

Extraction Process (X_E- Regression Coefficient=0.330)

- Earns qualitative components for reusable repository
- Is unnecessary, its usage will make reusability process complex
- Reusable Test is implemented during Extraction Process
- Reusable test checks software component’s reliability during
- Reusable test checks software component’s portability
- Reusable test checks software component’s generality
- Reusable test checks software component’s understandability

The Relation of independent variables with dependent variable can be shown graphically as follows:

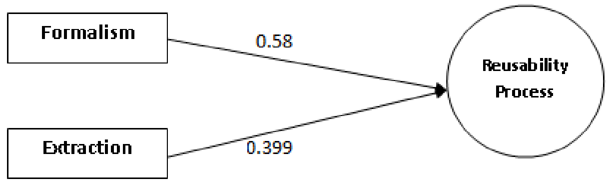

Figure 4. Significant Independent Variables Relation with Dependent Variable

Our findings indicate that to improve effectiveness of reusability methodology, formalism is essentially required. This formalism can be achieved by properly identifying, extracting, storing, classifying and deploying reusable components. Being a survey based study its results are obtained from different software development organizations. The important thing in mentioned results is that facts are describing current usage level and effect of reusability. These findings are also suggesting a trend of formalism in reusability methodology that can maximize its effectiveness.

6. Conclusion

This paper is a survey based study to measure effectiveness of reusability methodology and role of formalism in its effectiveness. To obtain formalism in reusability methodology its process is divided into three stages. These stages are Extraction, Storage and Deployment of Reusable components. But this survey is only limited to Extraction Stage. This survey consists of 10 sections and 42 factors, which are measuring reusability process and role of formalism in its effectiveness. Data analysis was made using Multiple Regression Technique. 12 independent variables out of 30 have significant effect on reusability process, other 17 factors are found to be insignificant.

As the results have shown formalism has positive effect on reusability methodology so in future we will apply formalism on remaining stages of reusability process. So it is concluded that after formalizing each activity of reusability process, effectiveness of reusability can be optimized.

Acknowledgements

We are highly appreciative to Mr. Ahmed Ali Lecturer in Commerce Department, University of Sargodha Bhakkar Campus for providing us help in statistical analysis of the hypothesis. His sincere and steady support contributed strongly to achieve our task effectively. We are also thankful to all organizations for the provision of data which made analysis possible.

References

[1] S. R. Kumar and P. S. Grover, "Managing Component Based System with Reusable Components", International Journal of Computer Science and Security, vol. 1, no. 2, **(2007)**.

[2] M. Farhan, A. Ahmed, M. Ramzan, S. R. Bashir, M. M. Iqbal and F. Ahmed, "Reusable Open Source Software Component's Life Cycle Management", International Journal of Multidisciplinary Science and Engineering, vol. 3, no. 4, **(2012)**.

[3] M. Ramachandran, "Automated Improvement for Component Reuse", INFOCOMP Journal of Computer Science, vol. 4, **(2005)**.

[4] G. Singaravel, V. Palanisamy and A. Krishnan, "Overview Analysis of Reusability Metrics in Software Development for Risk Reduction", International Conference on Innovative Computing Technologies (ICICTI), Tamil Nadu, India, **(2010)** February12-13.

[5] W. J. Salamons and D. R. Wallace, "Quality Characteristics and Metrics for Reusable Software", Preliminary Report, U.S. Department of Commerce, **(1994)**.

[6] K. K. Aggarwal, Y. Singh, A. Kaur and R. Malhotra, "Software Reuse Metrics for Object-Oriented Systems", Proceedings of the 2005 Third ACIS Int'l Conference on Software Engineering Research, Management and Applications (SERA'05).

[7] A. O. Bajeh, "A Novel Metric for Measuring The Readability of Software Source Code", Journal of Emerging Trends in Computing and Information Sciences, vol. 2, no. 10, **(2011)**.

[8] R. S. Pressman and Associates, Inc. Adaptable Process Model, http://www.rspa.com/apm/umtask06.html, 29-12-2012.