International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2014): 3.05

The Increase in Certified Software Reuse Decreases Defects Thereby Increasing Quality and Productivity

Ashwin Tomar[1], V. M. Thakare[2]

[1]Computer Science, MCA Department, Under Pune University, Pune, India
Siddhant Institute of Computer Application

[2]Computer Engineering & Science, Amravati University, Amravati, India, P.G. Dept. of Computer Science

Abstract: ***Software reuse is very important to speed the work of development, reduce cost and time. Software reuse is associated with requirement, architecture, code, components, documents in various phases of software development. In this paper independent and dependent variable are consider using regression technique. Their behavior is studied and relationship is established between them.***

Keywords: DRE – Defect removal efficiency, DP – defect potential, REQ-Requirement, DSN - Design, CDE - Coding, DOC-Document.

1. Introduction

Software reuse is the process of creating software system from existing software rather than building from scratch. Software is the process of implementing or updating software systems using existing software components. A software reuse process facilitates the increase of productivity, quality & reliability, and the decrease of costs and implementation time. An initial investment is required to start a software reuse process, but that investment pays for itself in a few reuses. So the development of a reuse process and repository improves quality after every reuse, minimizing the amount of development and reducing the risk of new projects that are based on repository knowledge.

A defect is nothing but a variance from the given specification, a hidden or coding error. Defects are undesirable, they cause increase in risk, revenue loss to the customer if they remain in the final product. DRE i.e *defect removal efficiency* refers to the percentage of total defects found and removed before software applications are delivered to customers. *Defect potential* refers to the total quantity of bugs or defects that will be found in five software artifacts: requirements, design, code, documents, and bad fixes or secondary defects [1].

A good data file from www.namookanalytics.com is reported by Dr Caper Jones is considered. The data and variables and their behavior is studied. The paper is arranged in following sequence as introduction, literature survey, methodology and data collection following by result, conclusion and reference.

2. Literature Survey

P. D. Patel [2] presents defect forecasting algorithms to analyze the defect using mining technique. Gunes K et.al[3], proposed a defect prediction model. Researcher K A Briki et.al[4], has worked on defect of code and suggested how to minimize the code defect to improve quality and thereby reduce development cost. B.H. Wu[5] worked hard for modelling of defect in software system. Dr Caper Jones worked on measuring area of software using metrics[6]. D.Y.Gattaiah et.al[7], improves Software Quality Assurance using bug tracking system which is an automated system that can be useful to employees and the managers in any functional organization. This Bug Tracking System gives the facility to define the tasks in the organization and also allows the managers to track the bugs spent by the employee for that particular task. A report generation facility is supported in BTS(Bug Tracking System) that allows the managers to analyze which those skills by employee are utilized and those which are not utilized. This tool can help managers for Bug estimation per project or application. This tool helps employees to document their bugs and analyze them.

3. Methodology

Software reuse is the use of engineering knowledge or artifacts from existing software components to build a new system. There are many work products that can be reused, for example source code, designs, specifications, architectures and documentation. Open source code can also be reused[8]. Code reuse is accomplished through the sharing of common classes and/or collections of functions, frameworks and procedures**.** Reusable software components can be simple like familiar push buttons, text fields list boxes, scrollbars, dialogs. A component is a object in the graphical representation of application that can interact with user. The user must access these components accurately and quickly, and be able to modify them if necessary. The component reuse is the reuse at code level.

There can be different level of reuse Code level components (modules, procedures, subroutines, libraries, etc) Entire applications, Analysis level products, Design level products. Code level reusable components can be in form of Class libraries, Function libraries, Design patterns, Framework Classes. Reused components have lower defect-density than non-reused ones [9]. Four STAGES like Requirement (REQ), Design (DSN), Coding (CDE), Document (DOC) are considered with abbreviations as shown in Table 1.

Table 1: Four Project Stages considered

Stage	*Description*	*Remark*
REQ	Requirement	It represents the short form
DSN	Design	It is the short form
CDE	Coding	CDE is abbreviation used
DOC	Document	DOC is short form of it

Table 2: Table showing Independent variable with its description

Independent Variable	*Description*
STAGE	Variable represents stage like requirement, design, coding, documents
CReuse	Variable represent certified reuse in percentage

Function points is a unit of measurement to express the amount of business functionality an information system (as a product) provides to a user. Function points measure software size. Larger the number of function point larger will be the size of software. The Table 3 shows defects per function points which are related to the different phases. The table 3 shows the defect potential circa 2012 for United States average [10].

Table 3: Software Defect Potentials

Phases with defects	*Function Points*
Requirements	defects per 1.00 per function point
Architecture	defects per 0.30 per function point
Design	defects per 1.25 per function point
Code	defects per 1.50 per function point
Document	defects per 0.60 per function point
Test case	defects per 0.75 per function point
Bad fix	defects per 0.35 per function point
TOTAL DEFECTS	defects per 5.75 per function point

There are different defect preventive methods like high quality component reuse, Quality function deployment (QFD), Root cause analysis, Six sigma, Clean room software development, Total quality management (TQM),Quality measurements, Quality Circles, Orthogonal defect analysis, Defect tracking tools, Static analysis, Formal design inspections, Formal code inspections, use certified components, checklist, reviews.

4. Data Collection and Application of Regression Technique

A good data file with 61 cases(0 to 60) of Software Risk Master™ Quality collected by Dr Caper Jones with details on Stage (requirement, design, code, document, bad fixes) was considered. The file has Independent variable like Certified Reuse in percentage, dependent variable like Defects(d). The excel file the operation on data was performed and relationship was established between percent reuse with various stages of defect [10].

Regression analysis is a statistical process for estimating the relationships among variables. Regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed. Regression analysis estimates the conditional expectation of the dependent variable given the independent variables - that is, the average value of the dependent variable when the independent variables are fixed. The estimation target is a function of the independent variables called the regression function. Regression analysis is widely used for prediction and forecasting. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. In restricted circumstances, regression analysis can be used to infer causal relationships between the independent and dependent variables[12].The regression technique is applied on the data file and after many operation on the data file, the relationship is established between defect, reuse and STAGES.

Table 4: Average defects at various stages as per reuse %

	0	15	20	25	30	40
Bad fixes	272.98	200.68	151.52	152.76	167.78	39.24
Code	1090.7	592.61	557.48	648.41	487.11	417.32
Design	1223.85	1070.72	890.69	842.02	915.65	376.99
Documents	563.41	483.36	380.55	369.53	401.08	116.53
Requirements	934.61	848.01	719.449	742.39	693.51	301.59

Table 5: Average defects at various stages as per reuse … continued from above Table

50	**60**	**75**	**80**	**85**	**90**	**93**
99.39	27.88	40.84	40	29.24	15.93	4.03
254.52	97.34	77	132.41	71.12	41.05	20.6
593.08	156.61	275.01	239.24	175.8	106.46	56.43
261.03	71.32	121.27	102.32	79.39	46.95	22.24
478.08	134.58	233.94	189.48	137.83	90.56	50.07

The linear and exponential graph is obtained from the data. The linear equation is written as

$y = mx + c$ and of exponential it is written as $y = m.e^{-x}$

Table 6: Finding relationship between Defect & Reuse

CMM	**Linear**		**Exponential**	
REUSE	***R Square***	***Equation***	***R Square***	***Equation***
REQ	$R^2 = 0.903$	$y = -9.597x + 916.7$	$R^2 = 0.865$	$y = 1260.e^{-0.02x}$
DESIGN	$R^2 = 0.899$	$y = -12.21x + 1155$	$R^2 = 0.859$	$y = 1603.e^{-0.02x}$
DOCU	$R^2 = 0.863$	$y = -5.443x + 509.8$	$R^2 = 0.832$	$y = 691.3e^{-0.02x}$
CODE	$R^2 = 0.882$	$y = -9.531x + 831.3$	$R^2 = 0.918$	$y = 1326e^{-0.03x}$
BAD Fixes	$R^2 = 0.841$	$y = -2.467x + 221.3$	$R^2 = 0.796$	$y = 323.3e^{-0.03x}$

5. Result

Except coding all graphs are linear. It is obvious that with increasing certified reuse, the occurrence of defects decreases. At most stages the decrease shows a linear trend, which is proportional to the extent of reuse. However in case of coding the reuse of codes comes into effect, where as in other cases reuse could be of documents, design patterns etc. When it comes to coding the effectiveness is exponential. Developers and source code writers benefit most by reuse.

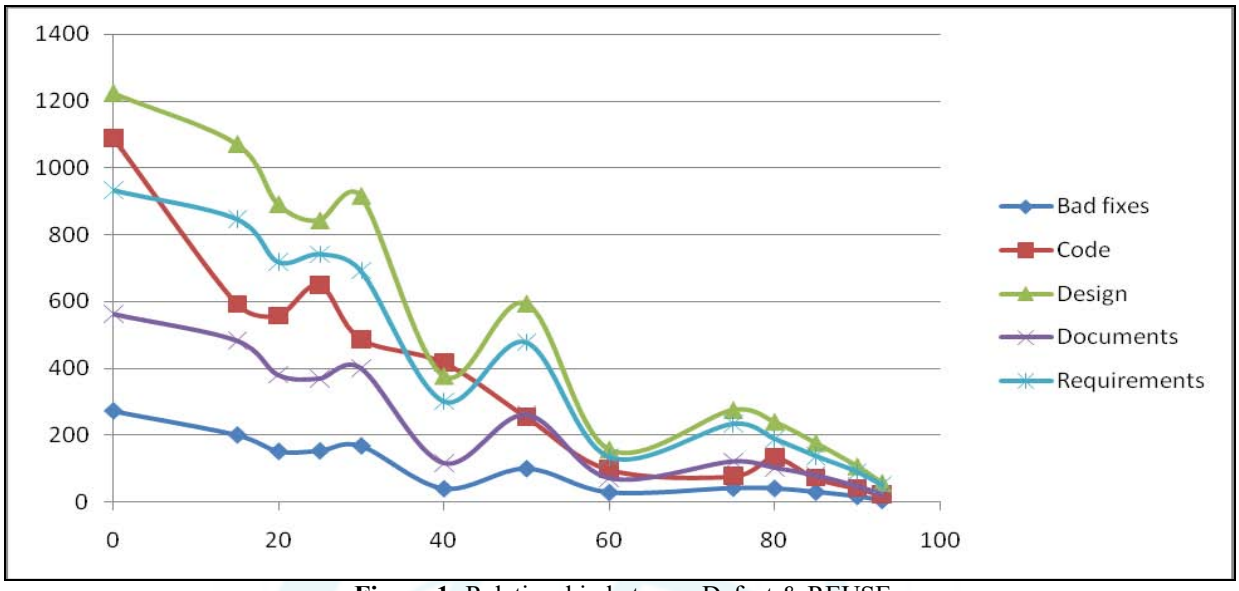

Figure 1: Relationship between Defect & REUSE

6. Conclusion

The relationship between percent certified reuse is established with defect in relation to various STAGES like Requirement (REQ), Design (DSN), Coding (CDE), Document (DOC), Bad Fixes. Except coding all graph shows linear trend. Increase in certified reuse decreases defect thereby increasing quality and productivity.

References

[1] Caper Jones, “Measuring Defect Potentials and Defect Removal Efficiency”, Software Productive Research, June, 2008
[2] P. D. Patel, “Defect Forecasting In Software System - Mining Approach,” IJECS, vol. 3, no.1, pp. 3763–3767, 2014.
[3] A. Günes¸ Koru Hongfang Liu, "Building Effective Defect Prediction Models in Practice", *IEEE SOFTWARE,* Vol 22, No 6, p.p 23-29, Nov - Dec 2005.
[4] K A Briski, Poonam Chitale, Valerie Hamilton, Allan Pratt, "Minimizing code defects to improve software quality and lower development costs", IBM, Development solutions White paper, Oct 2008.
[5] B. H. Wu, “Modeling defects in software systems,” 2011 IEEE *International Conference on Granular Computing,* pp. 739–744, Nov. 2011.
[6] Capers Jones, "Software Defect Origins and Removal Methods", Dec 28, 2012, www.Namcook.com.
[7] R. Kumar, D. Y. Gattaiah, S. Shahi, and T. A. Nagendra, “Improving Software Quality Assurance Using Bug Tracking System,” IJCSIT, vol. 4, No 3, pp. 492–497, 2013.
[8] Y. Tung, C. Chuang, and H. Shan, “A Framework of Code Reuse in Open Source Software,” 2014.
[9] P. Mohagheghi, R. Conradi, O. M. Killi, H. Schwarz, “An Empirical Study of Software Reuse vs Defect-Density and Stability,” Vol no. 4898, Proceedings of the 26th International Conference on Software Engineering (ICSE’04) 2004.
[10] Caper Jones," Software defect origin and removal methods", www.namcock.com
[11] Ashwin Tomar, V.M.Thakare, “The Study of models for Software Quality Assurance, reuse and predicting a customized model" 2015, Thesis.
[12] www.wikepedia.com.