Scientific Journal of Impact Factor (SJIF): 4.14

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 2, February -2016

The Selection of software models for reducing defects thereby increasing quality and productivity of software

[1]ASHWIN TOMAR, [2]V.M.THAKARE, [3]ACHARYA HARIDAS

[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 Allana Institute of Management Science, Pune University, India*

Abstract: *The models are very important while developing the software. The crucial part lies in implementing them while developing the software. Various different models are studied and classified into various rough categories like classical, modern and latest. The regression technique is used to find relationship between the categories of model and the number of defect predicted while developing the software. The types of model is associated with the number of defect present. Lesser the defect, better the quality and productivity.*

Keywords: *DRE – Defect removal efficiency, DP – defect potential, STAGES – are associated with requirements (REQ), design (DSN), coding(CDE), document (DOC).*

I. INTRODUCTION

Model is graphical, mathematical(symbolic), physical, or verbal representation or simplified version of a concept, phenomenon, relationship, structure, system, or an aspect of the real world. The objectives of a model include to facilitate understanding by eliminating unnecessary components, to aid in decision making by simulating 'what if' scenarios, to explain, control, and predict events on the basis of past observations[1].

While developing software defects are brought to the notice of project team by a process known as bug reporting. Defect reports are used to alert software programmers about the defect and gives them sufficient information to find root cause of problem and fix it. It provides information to technical writers and add test cases in the regression suite for next release[2].

In this paper the relationship between different models, approaches, methodologies considered while development and number of defect present in various STAGES like Requirement(REQ), Design(DSN), Coding(CDE), Document(DOC) is established based on data file collected by Dr Caper Jones. The paper is arranged in following sequence introduction, literature survey, models methodology defects and their relationship, data collection followed by result, conclusion, references and appendixes.

Literature Survey:

Dr Caper Jones worked on measuring of software using metrics[3]. Hazif A.Khan et.al[4] proposed a defect management process model & establish it in an organization to reduce the number of defects and produce a quality software product. The defects are detected, removed and prevented so that quality is increased. Models on quality assurance are proposed by different researchers. One such example is a quality assurance model for analysis phases by R. Ejaz, M.Nazmeen et.al[5]. Manju lata et.al[6] proposed a quality assurance model to optimize cost. B. H. Wu[7] proposed a defect prediction model. H.A.Khan[8] proposed a defect management process model in an organization to reduce the number of defects and produce a quality software product. Defect management process includes three levels which are defect detection, defect analysis and defect prevention to eliminate and mitigate the potential defects.

Models, Methodology, Defects and their Relationship:

There are different models, methodology like waterfall model, Iterative, Object-Oriented, Agile with scrum, Rational Unified Process (RUP),Model-based Certified reuse etc which are used for the development to produce high quality products. There are different defect preventive method 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.

Most failures in software products are due to errors in the requirements and design phases as high as 64 percent of total defect costs & 36% for coding / implementation phase. The Table 1 shows the percent of defect introduced in software development phases varies. Errors in software requirements and software design documents are more frequent than errors in the source code, as they could not be removed earlier the cost increases.

Table 1: Defects associated with Phase

Software Development Phases	**% of Defect introduced**
Requirement	20 %
Design	25%
Coding	35%
User Manuals	12 %
Bad Fixes	8 %

Requirement defects are severe types than other types of defects. **DRE** i.e defect removal efficiency refers to the percentage of total defects found and removed before software applications are delivered to customers. **Defect potentials** 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[9]. If software has Defect potential less than 2.5 % and DRE above 96% then the Quality in the software is considered as Best class. If DRE is below 60%, in such a case the software is of POOR quality. Requirement should be stable and there should not be requirement changes more that 2.5% [9].

There are different models, methodology, approaches which were classified into I(Classical), II (Modern), III(Latest) categories. These models are being used while developing software. The abbreviation used for some of them are described here and in appendix: FBP –flow based programming, MSF – Microsoft solution framework, EXP – experience, AWS – Agile with scrum, VM - V model, PAS – pseudo agile with scrum, TDD – Team driven development, MM – Multiple methods, EVO – Evolutionary development, MBD – Model based development, TSP – Team software process, RAD – Rapid application development, JAD – Joint application development, RUP – rational unified process, UML–Unified modelling language. ISO/IEC - International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), MDE - Model-driven engineering, QFD – Quality function deployment.

Table 2: Showing Classification of Methodologies based on above three categories (3 * 3)

Category	**Technologies / Approaches Family (Subcategory1) A**	**Subcategory 2 B**
Old, Structured oriented ***I CLASSICAL***	Waterfall. Cowboy, Merise, Prince2, Oliva Nova Model,	Unstructured
Intermediate *Object Oriented and Others* ***II MODERN***	RAD, JAD, TDD, Clean room, Spiral, Incremental, Iterative, Prototyping, Evolutionary, Mashup, RUP, UML,V-model, QFD, Prototyping, IE, TSP, ISO/IEC, AWS, PSP, Hybrid, Inspection, T-VEC, Six Sigma, Multiple methods, MDP (model based dev), MMBD, Flow based programming	Customized

Current, Latest **Resource**s & Domain Oriented, *Service Oriented* e.g. Cloud computing Enterprise and application architecture ***III LATEST***	Agile, Microsoft Sol framework, SOMA, PAS, MSF	Scrum, eXtreme programming, Lean, Kanban, USDP, FDP

Data Collection:

A good data file reported by Dr Caper Jones was taken as basis. Totally 61 cases (0 to 60) of Software Risk Master™ Quality with details on Stage (requirement, design, code, document, bad fixes). The data collected shows variables like Independent variable (Model Types), Dependent variable (Defects). The various abbreviation used for various stages like requirement, design, coding, document with their description are shown in Table 3.

Table: 3 The short form used for Project Stages

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

The Table: 4 shows Independent variable like models (MOD, MODL), Stages like requirement (REQ), design(DSN), coding(CDE), document(DOC).

Table 4: Showing Independent variable with its description

Independent Variable	Description
STAGE	Variable represents stage like requirement, design, coding, documents
MODL, MOD	Variable is used for different MODEL like Water fall, Agile

Various operations with regression technique was applied on data file and number of defects found are based on classes of Models (I, II, III) as shown in Table 5.

Table 5: Average defects at various stages as per types of Model

STAGES	I	II	III
Bad fixes	334.7298094	155.4947767	255.9616776
Code	1443.76819	593.6226938	806.6955924
Design	1375.288988	832.2164196	1136.676842
Documents	635.2570885	372.9749067	526.9999074
Requirements	1053.111705	639.6717186	917.9334381

The relationship is finded using regression technique between Types of Model and Defects.

Table 6: Finding relationship between defect & types of model

Model	Linear		Exponential	
	R Square	***Equation***	***R Square***	***Equation***
Bad fixes	$R^2 = 0.192$	y = -39.38x + 327.5	$R^2 = 0.118$	$y = 310.0e^{-0.13x}$
Document	$R^2 = 0.168$	y = -54.12x + 620	**$R^2 = 0.119$**	**$y = 602.4e^{-0.09x}$**
Code	$R^2 = 0.518$	y = -318.5x + 1585	$R^2 = 0.415$	$y = 1582.e^{-0.29x}$
REQ	$R^2 = 0.102$	y = -67.58x + 1005.	$R^2 = 0.071$	$y = 977.4e^{-0.06x}$
Design	$R^2 = 0.192$	y = -119.3x + 1353	$R^2 = 0.141$	$y = 1320.e^{-0.09x}$

Linear equation's graph shows constant rate of increase or decrease while in quadratic equation graph, there is constant addition. In case of exponential graph there is constant multiplication. The graph is plotted based on Table 5 and Table 6.

Result:

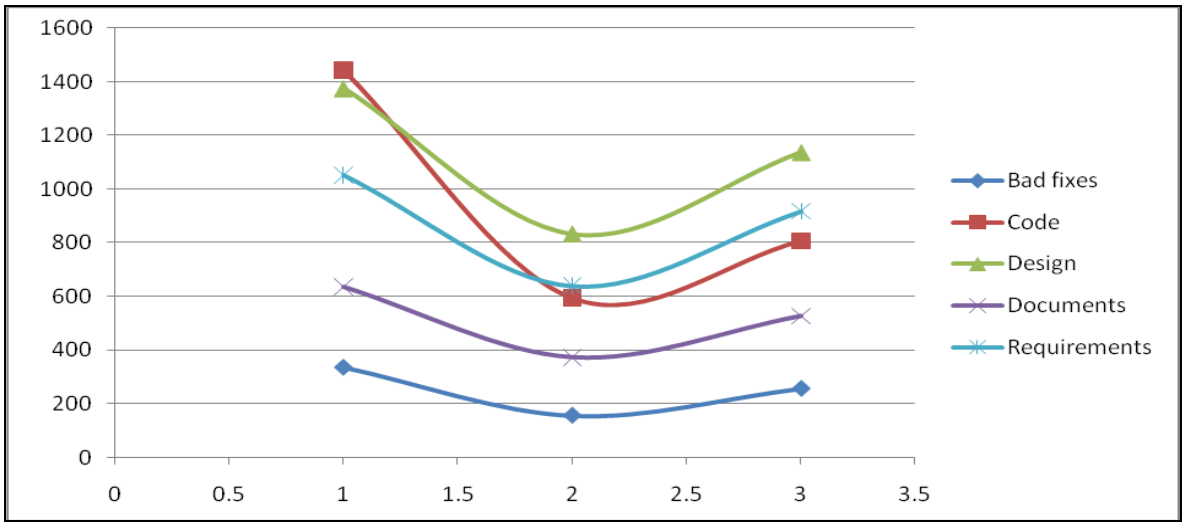

Fig 1: Curve Fitting for Defects of various Vs Model

As per caper Jones data all cases have linear exponential decay. Errors are more during requirement, designing and coding stages. With better use of modern models the errors reduces. Defect potential and defect delivered goes on decreasing from waterfall model to certified reuse methodology (i.e classical to modern and latest models) while DRE (defect removal efficiency) increases from top to bottom. The number of defects are associated with cost, time and quality. The lesser the number of defect indicates better quality.

CONCLUSION

The relationship between various models, methodology, approaches and number of defects present in various STAGES like Requirement(REQ), Design(DSN), Coding(CDE), Document(DOC) is established. Number of defects predicted are based on various model classified into I, II,III categories. The defects number present are associated with cost, time and quality. The lesser the number of defect indicates better the quality and less money loss. The type of model is associated with cost, time, improvement, quality and reduce rework. Based on it good or poor quality software can be rated.

REFERENCES

[1] www.businessdictionary.com/definition/model.html
[2] www.wikipedia.org/wiki/Bug_tracking_system; www.wikipedia.org / wiki / Defect_tracking; www.wikipedia.org/wiki/Software_bug;
[3] Capers Jones, "Software Defect Origins and Removal Methods", Dec 28, 2012, www.Namcook.com.
[4] H. A. Khan, “Establishing a Defect Management Process Model for Software Quality Improvement,” *IJFCC*, vol. 2, no. 6, 2013.
[5] R. Ejaz, M. Nazmeen, M. Zafar, “A quality assurance model for analysis phase,” *NSEC*,’ pp.1-4, 2010.
[6] M. Lata, R. Kumar “An Approach to Optimize the Cost of Software Quality Assurance Analysis,” IJCA, Vol. 5, no. 8, pp. 1–4, 2010.
[7] B. H. Wu, “Modeling defects in software systems,” ICGC, pp. 739–744, Nov. 2011.
[8] H. A. Khan, “Establishing a Defect Management Process Model for Software Quality Improvement,” IJFCC, vol. 2, no. 6, 2013.
[9] Caper Jones, “Measuring Defect Potentials and Defect Removal Efficiency”, Software Productive Research, June, 2008.
[10] Ashwin Tomar, V.M.Thakare, " The Study of models for Software Quality Assurance, reuse and predicting a customized model" 2015, Thesis.

APPENDIX

The details of various models are explained here.

1. **PSP** - The Personal Software Process is a structured software development process, uses a "disciplined, data-driven procedure". The PSP was created by Watts Humphrey to apply the underlying principles of the Software Engineering Institute's Capability Maturity Model to the software development practices of a single developer.
2. **USDP** - Unified Software Development Process is software design methodology, Booch and Rumbaugh introduced in 1999 the Unified Software Development Process (USDP) as a software engineering process.
3. **eXtreme** - is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements.
4. **Scrum** - Scrum is an iterative and incremental agile software development methodology for managing product development.
5. **Lean** - Lean software development (LSD) is a translation of lean manufacturing and lean IT principles and practices to the software development domain. Adapted from the Toyota Production System, a pro-lean subculture is emerging from within the Agile community.
6. **Kanban** - Kanban is a method for managing knowledge work with an emphasis on just-in-time delivery while not overloading the team members.
7. PAS - Performance Aware Software Development.
8. MSF – Microsoft solution framework(MSF) is a set of principles, models, disciplines, concepts, and guidelines for delivering information technology solutions from Microsoft. MSF is not limited to developing applications only, it is also applicable to other IT projects like deployment, networking or infrastructure projects. MSF does not force the developer to use a specific methodology (Waterfall, Agile) but lets them decide what methodology to use.
9. MBD/MMBD – Model based design / development
10. QFD – Quality function development
11. RAD – Rapid application development
12. JAD - Joint application development
13. MM - Multiple methods
14. **ISO/IEC** is a standardization subcommittee of the Joint Technical Committee ISO/IEC JTC 1 of the International Organization for Standardization (ISO) and the International Electro-technical Commission (IEC), that develops and facilitates standards within the field of engineering of software products and systems. The international secretariat of ISO/IEC JTC 1/SC 7 is the Standards Council of Canada (SCC) located in Canada.
15. Model-driven engineering (MDE) is a software development methodology which focuses on creating and exploiting domain models, which are conceptual models. The MDE approach is meant to increase productivity by maximizing compatibility between systems (via reuse of standardized models), simplifying the process of design (via models of recurring design patterns in the application domain), and promoting communication between individuals and teams working on the system (via a standardization of the terminology and the best practices used in the application domain).
16. **SOMA** - Service-oriented modeling is the discipline of modeling business and software systems, for the purpose of designing and specifying service-oriented business systems within a variety of architectural styles, such as enterprise architecture, application architecture, service-oriented architecture, and cloud computing
17. **USDP-Unified Software Development Process or Unified Process, I**s a popular iterative and incremental software development process framework. The best-known and extensively documented refinement of the Unified Process is the Rational Unified Process (RUP). Other examples are OpenUP and Agile Unified Process.
18. **AWS** - Agile with scrum & **FDP** – Fuzzy dynamic programming.

