Software Quality Measurement using complexity Analysis of Various Software Engineering phases

Neha Bharani, Dr. Abhay Kothari

Article Info
Volume 83
Page Number: 9437 - 9447
Publication Issue:
May - June 2020

Article History
***Article Received**: 19 November 2019*
***Revised**: 27 January 2020*
***Accepted**: 24 February 2020*
***Publication**: 18 May 2020*

Abstract:
The development of software has taken in significant proportions in regard to other aspects of system development. We have identified the common Complexity we face to perform this work, we've Selected 10 varieties of software (Such as AI, Mobile app….)and four development phases (Planning, Evaluation ,Design, Implementation Testing and Maintenance) and attempting to determine the consequences of software type on the problem offered during each software engineering phase. We are taking several solved problems within each software type to assess the complexity level and together with this we are attempting to administer the logic behind a specific difficulty level assessment. Foresee and inspection attribute already in product support some kind of upgrade of Product class. Software quality measures how well software is to be designed, and the way well the software conforms thereto design. This sort of labor are going to be useful for industry altogether their software engineering related work incurred while software development, particularly medium and enormous size projects.

***Keywords**:* Automation, Supervise, idiomatic expressions, identifying observed, guiding, proof-based work, annotative model, applied work, structured and unstructured enquiries, systematic, validity.

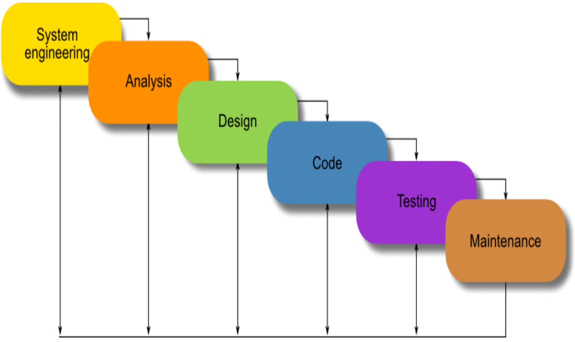
Introduction:

Software engineering is Associate with the development of product creating use of well-defined scientific principles ways, during which the very best results of code package take place. The process of developing a product package engineering principles and strategies is noted as package evolution.

The kinds of matter influence people with whom we usually work which is generally many sided and tough to explain, they are tough issues. With this term difficult issues are issues which are far away the reach of any organization to take it and give the response back in which all the issues are not fully resolved. It involves the initial growth of product and work on it till the actual software is developed. [4]

II SOFTWARE ENGINEERING ACTIVITIES

- Customer Communication
- Planning
- Modeling (Evaluation , Design)
- Construction(Coding , Testing)
- Deployment



A. *Planning:* - In this platform, project goals are descided and great plan for the calculate project is confirmed .Planning is very necessary phase. At lower level software are less complex, at managerial level software are complex and at higher level planning are more complex the primary activities involved within the planning phase are as follows:

Identification of the system for development

- Feasibility assessment
- Creation of project plan

B. *Evaluation*:-In this we will see the needs of client and according to that software and project aims will be change into some useful given functions which system wants to developed

The three steps include in this work is:

- Collect business need
- Creating process diagrams
- Deep Knowledge

C. *Design:*
In this, we Make a rough design layout of overall system what exactly the need of client is and show the details which describe the exact details and functions of the need. This activity involves all design documentation.

D. *Coding:* - Objective of the coding phase is to rework the look of a system into code in a very application-oriented language so to unit test this code. Good software system development organizations ordinarily need their programmers to stay to some well-defined and normal variety of commitment.

E. *Testing:* -Once product develop than testing take place just to check whether the work which is done is accurate or not. Which works for good standard of product?
Which is done in a very proper terms [2].the product and system which is developed is in good working condition or not properly doing it in order to verify the system and make sure it works is correct.

III UTILITY BASED ON SOFTWARE SYSTEM QUALITY

In absence of the utilization based on product system quality. It should more tough along with time overwhelming just figure out the work wherever risk and price flow a lot of significantly, continuously.

Software quality allows project groups and computer authority to urge before the matter and stop using quality from base. Once measure quality. It is additionally vital to possess Associate in nursing correct, recurrent terms of quality grade, consonant across the system steps of the appliance folder to give standard for go on with assessment. A powerful computer code quality measuring code gives a company with the prospect to:

- Increase program Status
- Scale back continuity Answer
- Heighten Productivity
- Increase strength
- Meet design Standards

IV RESEARCH METHODOLOGY

Research Methodology presentation to software development methodology, various growth perspectives have seen in software growth and change a lot according to latest trends which is implemented through software growth point of view. It also includes so many steps in the growth of product and all the steps are following and provide the best result.so much non-identical software representation for achieving the best and better result.so many software programs fails at the time of development.
The volume of peoples mind for evaluating and resolving tough problems in some small modules and then solve that modules in easiest an affordable Answers.

AI is modifying all work, and product growth isn’t a special case as well as machine learning skills are accustomed speed up the quality of product growth, they show a fully different pattern of latest trends.

Expanding a system code needs you to identify earlier absolutely which we provide computer to undertake as well as enhance all the models of latest trends.

So many work and results are very tough to show system in an exceedingly hard format based. Although tasks are apparently easy to know about whether a photograph or reel on the net is exceed the attain of Conventional software growth [4].

V MACHINE INTELLECT

Capability of computing systems to be told by withdrawing design from information is thought even as Machine Learning. Satisfaction and advantages of ML are:

- Strong Action
- Good deciding & Projection
- Fast Action
- Correct
- Economic Data dealing
- Affordable
- Examine compound information

In this work we have a situation where we compute the complexity of software of various types, and use this to predict the effects in terms of effort and difficulties to be face while conducting different phase (examination, design…) activities for development of the software. For this prediction use of some AI/ML techniques are suitable. This is going to help out the preparations for s/w development in various scenarios such as industry, research etc. Some features, techniques, algorithms, and applications of machine learning found suitable for our work are presented below

Features of Machine Learning

- Machine Learning is computing-intensive and usually requires an oversized amount of coaching data.
- It involves repetitive training to enhance the educational and higher cognitive process of algorithms.
- As more data gets added, Machine Learning training is automated for learning new data patterns and adapting its algorithm.

Example: Learning from new spam words or new speech (also called incremental learning)

Machine Learning Techniques are:

- Classification
- Categorization
- Clustering
- Trend Examination
- Anomaly detection
- Visualization
- Decision making

A .Machine Learning Algorithms

Machine Learning can learn from marked data or unmarked data. Machine Learning algorithms involving unlabeled data, or unsupervised learning, are more complicated than those with the labeled data or supervised learning. Machine Learning algorithms are wont to make decisions in subjective areas yet.

Examples:

Logistic Regression are often accustomed predict which party will win at the ballots. Naïve Bayes algorithm can separate valid emails from spam.

B. Applications of Machine Learning

Some of the applications of Machine learning mentioned below.

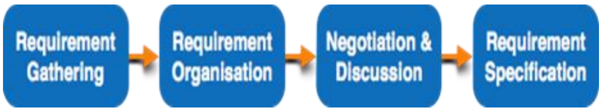
- Image Processing
- Robotics
- Data Mining
- Video Games
- Healthcare

The merchandise must be designed in such some way that it optimally performs the most task or function that it's purchased by a buyer.

- Repair ability
- Reliability
- Aesthetics
- Durability
- Simplicity
- Compact

VI REQUIREMENT INPUT METHOD

Requirement input method are pictured mistreatment the subsequent diagram [4].



A. Needs gathering - The developers talk to the consumer and finish users and apprehend their expectations from the package.

B. Organizing needs - The developers order and organize the requirements thus as of importance, urgency and convenience.

C. Negotiation & discussion - If needs area unit ambiguous or there are a unit some conflicts in needs of various stakeholders, if it is then negotiated and mentioned with stakeholders. Needs might then be prioritized and fairly compromised.

Documentation - All formal & informal, useful and non- functional desires area unit documented and created on the market for next method.

Functional Necessities

Needs that are associated with down to earth side of programming assortment fall under this class. They layout capacities and common sense inside and from the bundle.

Models –

- Client should have the option to mail any report back to the executives.
- Every Users will be partitioned into various groups and groups will be given separate work
- It is important to keep business rules and ideas.
- Software bundle is created keeping descending similarity flawless.
-

A. Non-Functional Specifications

Specifications which do not tend to be correlated with the software's technical component fall into this grouping. They tacit or anticipated software features which users presume.

Non-functional specifications include-

- Protection
- Recording
- Documentation
- Efficiency
- Expense
- Scalability
- Versatility
- Recovery from disasters

VII SOFTWARE STYLE

Software style could be a method to remodel user needs into some appropriate kind that helps the software engineer in package cryptography and implementation.

For assessing user needs, Associate in SRS (Software demand Specification) document is formed whereas for cryptography and implementation, there's a necessity of a lot of specific and elaborate needs in package terms. The result of this process will directly be used into application in programming languages.

Software manner is that the initiative in SDLC (Software style Life Cycle), that moves the cluster from downside domain to answer domain. It tries to specify the way to fulfill the needs mentioned in SRS.

B. *Software Style Levels*

Software style yields 3 levels of results:

- Architectural style - The order style is that the most noteworthy conceptualadaptation of the framework. It distinguishes the bundle as a framework with a few components associating with each other.
- High-level Design-The elevated level style breaks the 'single element different part' build of control style into less-preoccupied read of sub-frameworks and modules and portrays their association with each other. Elevated level style centers around anyway the frameworks together with the entirety of its components are frequently implemented in kinds of modules. It recognizes standard structure of each sub-framework and their connection and collaboration among each other.
- Careful Design-Detailed style manages the execution a piece of what's viewed as a framework and its sub-frameworks inside the past 2 styles. It's a great deal of cautious towards modules and their executions. It characterizes consistent structure of each module and their interfaces to talk with various modules.

C. *Software Necessities*

It offer with the suggests that to outline and describe the aim, value, and scope of software based comes that are undertaken to handle and satisfy some perceived would like. Over the years, several definitions are confined for what a package demand actually should be, and what makes a "good" demand, therefore there aren't any new revelations to be found here, no new rocket science, some commonsense thoughts.

Software necessities can facilitate United States address and answer these (and other) queries. Package necessities offer an outline, a codification, a specification of a package based mostly answer to be enforced to handle and satisfy a perceived would like. They supply descriptions of however the general and underlying system ought to behave, and the way well, at varied interaction levels: user, package and hardware. They'll characterize system attributes or properties and that they is also expressed in terms of rules and constraints.

D. *Functional Requirements*

The client level mastery must be depicted regarding what the designer is intended to make. This assignment tumbles to deliberate necessities: they too offer the "what" data for the bundle venture, anyway in cautious terms pertinent to an engineer. For example, an intentional interest would perhaps be: "The framework will require n favoring chief to choose one by and large 3 endorsement call alternatives: Approve, Disapprove, Defer." the essential crowds for deliberate necessities region unit designers and analyzers. Most strategies share a mix of the consequent phases of programming advancement:

- Work on issues
- Market Survey
- Gathering requirements for the proposed business solution
- Come up with a motivation or structure for the product based outcome.
- Execution(coding) of the product
- Check the software
- Formation
- Continuity and error fixing

VIII SCORING OF DIFFICULTY LEVELS

A. Processing levels:

- Data processing, report generation (one unit/function 1 point)

- Image processing (2 points per function)
- Data analysis (2 points per function)
- Solution architecting, information architecting (5 points per instance)
- Data Mining, pattern recognition (2 points per function)
- Data modeling (5 points per unit)
- Data Integration, visualization (2 points per instance)
- Knowledge processing (3 points per instance)
- Inference, knowledge representation, interpretation, reasoning (3 points per function)

We will find these words in each requirement and assess its difficulty/processing level.

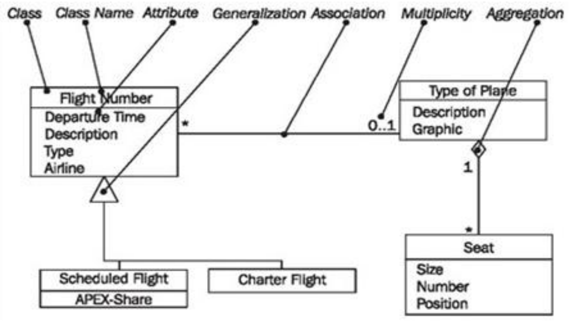
Scoring will be based on—requirements difficulty level, class diagram difficulty level, and code complexity

Requirement difficulty scoring: 1 point for each functional requirement 2 points for each non-functional requirements, for each functional requirement there will be a additional point of .1 for each level—for example any requirement falling in 9th level the functional requirement will weigh 1 + 0.9 = 1.9

Class diagram difficulty level: It is collected of three varieties of link, relation, induction, and gathering, through which models are more complex to understand [16].We are providing seen results which gives idea of logic to allow an aim of compound and continuity.

B. Scoring for Class diagram

- Class diagrams-number of classes, 1 point each
- Associations-- 1 point for per association
- Generalization 1 point for generalization
- Aggregation 1 point for aggregation
- Inheritance 1 point for inheritance



1 point for Class diagram 5 classes and 3 relationships and every relationship amounts for 2 points

Association

In various databases association rules are those If-then statements which assist to suggest the prospect of relations in data module among large data sets in different sorts of directory. It has so many requests and is generally reach to help in finding relation in valid data or in preventive datasets.

Generalization

Normally arithmetical data find out is worried along with approximate properties of main data division, instead of resources which are different to the countable data set at hand. Indeed, an oversized body of conceptual and inquiry was evolve for fixed abstraction in an exceedingly form of positions.

C. Processing Level Complexities Elaboration and Scoring

Image processing

A way to perform operations in such a way that normal picture converted into digitally equipped format so that we can extract useful data in an easy way. Process include an image as an input and output may be an image ,real to the format having some part of original image.it include automation ,agricultural and other healthcare ,Applications also.

Data anatomy

Process of finding out information by logical cause to check every module of the information each is given.

Such type of work is is simply numerous way which needs to be done when arrange a test.

Information from different origin is collect and examines just to find out the result.so many different source of particular data scanning ways from which it involves data action, visualization and observation just to resolve tough issues.

Data Mining and pattern recognition

Data mining is that the process of working on secret design of facts in step with various prospect for grouping into valuable information, that is gathered and build in same areas, like data warehouses, for better work. Data mining is called data finding and command discovery.

The major points elaborate in a data mining work are:

- Withdraw and convert information into a knowledge store.
- Gathered and control data in exceedingly several aspects databases
- Supply data entry to business workers using software.
- Present work data in predictable forms, like graphs.

Data modeling

Data pattern shows at the time of requirement of companies could be a very strong expression of the company's business needs. This data design is that the guide utilized by logical and technical requirements within the design and performance of a database.

Data design helps for various functions, from top level abstract structure to physical information structure.

Data modeling search data-oriented structures and identifies organization varieties. It may be in contrast
to category modeling, wherever categories area known.

Data integration

Data integration is the process of collecting information from different sources specified it is examine as a full to supply knowledge which is larger than the addition of its parts. Particularly, so many countless industrial challenges regarding the expandable of data-concatenation systems, moreover as tough issues regarding the character of the data itself and also the means through information could also be realized by humans.

Knowledge Processing

Knowledge presentation (KR) is that the work of how information and truth of the planet is present, and what styles of work can perform through this data.

Mainly KR query has to consider: representational acceptability representational standard, calculation cost of related reasoning, or uncertain information.

Data Interpretation.

Get the valuable and useful data from the huge amount of information whether the data is in any form hence requirements of explanation is very important.

- Data processing, report generation (one unit/function 1 point)
- Image processing (2 points per function)
- Data analysis(2 points per function)

- Solution architecting, information architecting(5 points per instance)
- Data Mining, pattern recognition(2 points per function)
- Data modeling (5 points per unit)
- Data Integration, visualization (2 points per instance)
- Knowledge processing (3 points per instance)
- Inference, knowledge representation, interpretation, reasoning (3 points per function)

Requirement examples for each processing level:

- Generate Mark sheets.
- Find out cancers tissues form given image/ enlarge the greyish part of the image
- Analyze the data statistically.
- Develop good software architecture to make software changeable.
- Find the pattern of goods purchased from departmental store.
- Make logical and physical data models of the data given.
- Integrate the structured and unstructured data of patients/employees.
- Perform knowledge processing of the given facts.
- Write a function to infer the intention of customer from the feedback data.

Scoring Example:

R1 Find the pattern of goods purchased from departmental store. (2 point)

R2 Make logical and physical data models of the data given. (5 X 2 = 10)

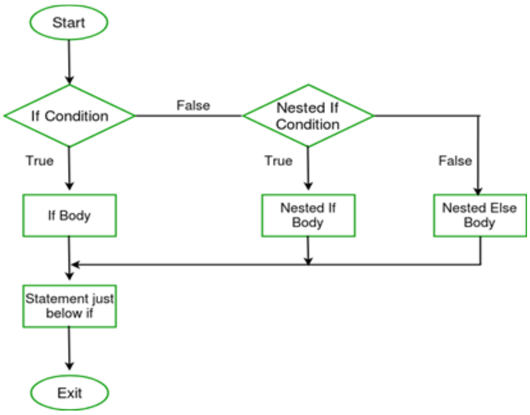
TOTAL = 12

E. Scoring of Program complexity

- Libraries
- Inputs statements
- Conditions
- Compound condition

Syntax of nested if...else statement.

```
else if(testExpression2)
{
  // statement (a)
}
else if (test Expression 3)
{
  // statement(c)
}
else
{
  // statement (f)
}
```



```
#include<stdio.h>
void main ()
{
   double na1, na2, na3;

Printf ("Enter three numbers: ");
Scanf ("%lf %lf %lf", &na1, &na2, &na3);

   If (na1>=na2)
   {
      If (na1>=na3)
Printf ("%.2lf largest number.", na1);
      else
Printf ("%.2 the largest number.", n3);
   }
   else
   {
      if (n2>=n3)
```

```
  printf("%.2lf largest number.", na2);
        else
  printf("%.2lf largest number.",na3);
     }
```

E .Program Scoring

2 points for conditions 1 point for library functions 1 point for subroutine functions, procedure and 2 for compound conditions.

For example If (a<b)&&(c<b) has 2 conditions x 2 = 4 points

The above mentioned program

3 conditions x 2 = 6

1 library function x 1

Program complexity = 7 (as per our method)

Mobile Application

In software advancement basic work done very soon while tough problems take maximum time to complete.

The most important two areas of discussion while judging the complicacy of mobile applications.

Structure and Attribute

The structure of a mobile application involves the data and

Make some changes in data so that it will be more helpful while using it.

Structure

In so many task user go through with various types of information for different task.

Attribute

Most of the features of a mobile application are designed for the target user: what should the user be ready to do with the applying a number of the features are oriented towards the owner (or creator) of the application rather than the tip user. For the sake of brevity, both are included within the intuitively Average Time Complexity refers to the everyday behavior of the algorithm averaged over all possible inputs.

VIII SOFTWARE COMPLEXITY

The software should have existed functions of calculating different network measures such as shortest paths, geodesic paths, between centrality and other centralities (Katz, eigenvector, closeness etc.) dynamically. Additional edges from Python embody quick prototyping, simple to show, and multi-platform it may be troublesome and time intense to work out the subject field hotspots wherever risk and value emanates [16].

A strong code complexness measure program provides a corporation with following points.

- Enhance Code Status
- Shorten Maintenance value
- Increase Production
- Expand strength
- Reach design Levels

The reality is that most enterprise software is bad because:

The Company who is writing it is not very good at UX/UI design. Decisions at large software development organizations are often made by committee, where when presented with options for option A or option B, people will often split the difference and do both

Large customers often think they are "special", and make demands that are themselves generated by committee, or not well-thought out, and force the company to implement them.

Measures what proportion management flow exists during a program - for instance, in RPG, operation codes like so many conditions and additional order syntax tougher to know, thus measurement the extent of complexness present what proportion must be managed. System is often advanced, and has interactions of outside Component. That will really increase the complexness of the codebase on the far side live. In every case difficulty level is different and the result is deceptive. It is necessary therefore, reside coupling and bond of the system in the syntax dependable moreover so as to urge the actual system

Artificial Intelligence

Artificial Intelligence is also can be called as intelligent systems as they can make automatic systems that can be tough as well ,such system is having thinking capability like human brain thinks .such system can answer whatever question asked to it. Whether question is exactly same or different such system has capability to absorb modifications by various information directly or indirectly from the systems. It always learn from the information it contains does not rely very much on the data it consists.

CONCLUSION

Software quality measures however well software system is to be designed, and the way well the software system conforms to it style. This type of work will be useful for industry in all their software engineering related work incurred while software development, particularly medium and large size projects.

AI is obtaining approved at a sooner pace; guiding the way we are, link and work on client incidents. Research work we are going to undertake In Artificial Intelligence type to type complexity and we will also find out the complexity of subtype with in the work.In data processing a conclude structure is a program system that gives the results from various resources with so many new syntax based skills. Workout structure defines a key point in the execution of artificial intelligence and command - build structure.

Objective reasoning is the top most term for three non-identical types of reasoning. We have gone through the mobile application software which has the more complexity and we have explored the level of complexity in mobile application and they are determined by entities and features. We also studied software of different types and explored their levels of complexity

We have used various phases and formed ways of scoring their complexity. Here we have used requirements, class diagrams, and the programs for scoring purposes. Once we calculate the difficulty levels, this can be used to estimate the recourses of software development. This work we shall undertake as our forth coming work.

REFERENCES

1. J.P. Hudepohl ; S.J. Aud ; T.M. Khoshgoftaar ; E.B. Allen ; J. Mayrand software metrics and models on the desktop, Published in: IEEE Software Volume: 13 , Issue: 5 , Sep 1996).
2. Continuously Assessing and Improving Software Quality With Software Analytics Tools Andreas Jedlitschka ; Xavier French ; Lidia Lopez , Date of Publication:17 May 2019 IEEE Access (Volume: 7).
3. Bhuvan Unhelkar ,San Murugesan,The Enterprise Mobile Applications Development Framework, Date of Publication: 29 March 2010 Published in: IT Professional (Volume: 12 , Issue: 3 , May-June 2010).
4. Jiujiu Yu Research Process on Software Development , University, Hefei 230601, Model College of Computer Engineering, Anhui SanLian, China. [2018]
5. Wei; Zhang YananResearch on Software Metrics and Software Complexity Metrics. ,[2009/01/01]
6. Rong School of Software School of Information, Yunnan University of Finance and Economics, Kunming 650221, 30 1 April 2015 / Published: 8 April 2015.
7. A new method for measurement and reduction of software complexity, Yindun Shi; Shiyi Xu, Published in: Tsinghua Science and Technology (Volume: 12, issue: 21, July 2007).
8. T.J McCabe A Complexity Measure Published in: IEEE Transactions on Software Engineering Volume 2 Issue 4, July 1976.
9. Huan Li Dynamic analysis of Object-Oriented software complexity, 2nd International

Conference on Consumer Electronics, Communications and Networks (CECNet). Published in: 2012

10. Harmeet Kaur Software Complexity Measurement: A Critical work Punjab Technical University Jalandhar, Punjab, India (IJEACS), Volume: 01, Issue: 01, November 2016 ISBN: 978-0-9957075-0-4.
11. Timothy C. Lethbridge Software Engineering Cite as aSoftware Engineers: Data Collection Techniques for Software Field Studies. July 2005, Volume 10, Issue 3, pp 311–341|
12. Minimol Anil ISSN: 2278 – 1323 International Journal of Advanced Research in Computer Engineering (IJARCET) Analysis of Software Complexity Using Object Oriented Design Metrics, Faculty of Computer Studies Arab Open University, Kingdom of Bahrain. Volume 5, Issue 10, October 2016
13. Simmi K. Ratan, Tanu Anand, and John Ratan Formulation of Research Question 2019 Jan-Mar.
14. Complex Problem Solving: What It Is and What It Is Not by Dietrich Dörner1 and Joachim Funke2,Published online 2017 Jul 11.
15. Measuring the complexity of class diagrams in reverse engineering,Frederick T. Sheldon ,Hong Chung irst published:21 September 2006
16. Wuli, Shili, Renli(WSR) Article Research and Measurement of Software Complexity Based on and Information Entropy [software complexity systematically and points out existing problems in current research. [2015/04/01]