SOFTWARE METRIC TRENDS AND EVOLUTION

B Venkata Ramana, Dr. G. Narasimha

ABSTRACT

Definition - Software Engineering encompasses a process, methods for managing and engineering software and tools. The role of software has undergone significant change over the past half century. From card readers to scanner, from simple equation to artificial intelligence, kilobytes to terabytes, CPU performance from 1 MHz to 6 GHz, 8 bit to 128 bit operating systems. The evolution happened in terms of space, complexity, quality and ease. Legacy applications are attributed with poor quality later with modern applications it’s eradicated. In fact the need for the evolution may even become obvious even before the new system is deployed. With evolving software, the metrics also evolved to measure the quality, not just in terms of documentation but in availability, reliability and robustness of the applications. Process and product measures have been defined to measure the quality of the engineered/developed product. The quality models and industrial standards – Six Sigma, SEI CMMI, ITIL, ISO, PMBOK, Prince2 and other, have changed the estate of software process in the IT world. Each of these help in improving the software development process. In this paper we analyze the metric evolution and the impact it has on software industry. Agile modeling is the current customer sought after model where the metrics are still evolving to suit the customer and market needs.

Keywords: Software Metrics, Software Evolution, Quality Standards, Metrics Trend, Object Oriented Metrics, Agile process.

I. INTRODUCTION

The concept of software quality and the efforts to understand the measurable quantities and measure them in terms of quality factors and quality criteria. A metric is a quantitative measure of degree to which a system, component or process possesses a given attribute. Metrics are useful for cost and schedule future projects, to establish productivity trend over time, improve software quality, anticipate and reduce future maintenance needs. Metrics are generally classified under Products, Processes and resources. Goodman defines software metrics as [1]: "The continuous application of measurement-based techniques to the software development process and its products to supply meaningful and timely management information, together with the use of those techniques to improve that process and its products". The culture of
Organization also serves as a key differentiator between successful ones and the laggards. Again when teams are considered more important than individuals then it’s the system that drive the functions and individuals absence and indispensability is ruled out. In this paper, the focus is on the metric trends, the process models and the quality improvements and the quality standards to meet the increasing demand.

II. METRIC TRENDS

Software process is more than a framework of tasks which is needed to build a high quality products. The process refines itself to software engineering once it starts using the technical methods and automation tools. IEEE defines, a process as “a sequence of steps performed for a given purpose” [2]. Software development life cycle SDLC models, describe the software process structures. Process metrics are defined for SDLCs, which include the activities, methods, and standards to use. The use of software process metrics has enabled some organizations to much more effectively understand and control their software development process [3]. Process metrics can be categorized based on the stages in SDLC. These metrics include – feasibility metrics, requirements metrics, design metrics, code related metrics, testing metrics. All these are used by management to derive new metrics to check the health of the project.

A. FEASIBILITY AND REQUIREMENT METRICS

Feasibility studies are conducted to understand if the project goal can be accomplished. There can be various feasibility studies - Technical, Economic, Legal, Operational and Scheduling. Organization do check for these metrics while bidding for projects. These have become a new set of metric by marketing and finance teams before they bid for a project. These metrics include

- IRR - Internal Rate of Return, > 10%, the higher the better.
- NPV – Net Present Value. > 0, the higher the better
- ROI – Return On Investment. Generally >12%

Requirement engineering process starts with feasibility study, elicitation and analysis, validation and management. The cost of fixing an error early is easy than fixing at later stages in SDLC. The metrics include

- Size metrics – LOC of FPP as software evolved, Use Cases are used.
- Traceability metrics
- Completeness metrics
- Volatility metrics

B. DESIGN METRICS

Design metrics are part of the product metrics, which are collected during the design phase in the SDLC. With the new software evolving new design metrics are evolving depending on the processes and tools used to design the software product. These metrics include [4]

- Structural complexity
- Data complexity
- System complexity

With the advent of Object Oriented modeling, new metrics evolved. Below are few, which are categorized based on the OO paradigm.

- Information Hiding
- Inheritance
- Polymorphism

The next trend in evolution is COTS – Commercial-of-the-shelf, resulted in the next set of metrics as below. Components have been developed for reuse and finally the COTS.

- Cohesion
- Coupling
• Complexity like cyclomatic complexity.

C. SIZE RELATED METRICS

The implementation, referred generally as coding, is the next step where the design is put forth for development. These include conventional size oriented metrics – KLOC – Kilo Lines Of Code, FP – Function Point. These were the units (KLOC, FP) to measure the complexity of code. In 1970s KLOC is used to measure the size of the system and as an anchor to estimate cost and schedule of the application. Typical metrics are below
• Errors/KLOC or Man Months/KLOC
• Defects/KLOC
• Cost/KLOC

Function Point metric in 1980s was later proposed to effectively measure the functionality being delivered and used for cost and schedule estimation. The technique of functional modelling is used to model the relationship between the transactions and the complete application. The FP is measured using five components – External Inputs, External Outputs, External Inquiry, Internal Logical Files and External Interface Files. Understanding the software size is the key to understanding both productivity and quality. Few FP metrics include FP/work month, Defects per FP.

The Object Oriented related metrics are addressed in the subsequent section IV.A.

There are other metrics that check the program complexity, purity ratio, McCabe’s Complexity (control flow representation) measures, McClure Complexity and many more. These measure the control flow of the program/application.

D. TESTING METRICS

Testing gets compromised due to delay in the initial phases and the duration gets reduced to meet marketing needs. Waterfall model symptoms include late shoe-horning of non-optimal fixes, with no time to redesign kind of graph, finally delivering a very fragile, unmaintainable product with overhead costs [6]. For improving the product quality and controlling the project, later models and organizations have come-up with a set of test-related metrics to allow better control and facilitate consistent improvement. These include
• Unit Test cases Planned/executed/Failed
• Bugs closure per unit of time
• Rate of Defect injection
• Defect Removal Efficiency

E. TEAM BEHAVIOURAL METRICS

People, one among the resources metrics and one of the 4Ps of software management, are the key drivers of quality. New process models (PSP, TSP) [7] evolved to improve the quality of products by considering software engineer’s into focus.

PSP – Personal Software process, suggests methods, measures and templates towards right track of quality (in order to change the ineffective personal process). Later the lessons learnt in PSP are introduced in TSP – Team Software Process. TSP being self-directed teams to direct and plan the assigned tasks effectively. In PSP, the templates are used to measure the efficiently of self individually and improve on error reductions. Metrics are defined by individual or team based on the model chosen to track the quality and software development progress.

F. OTHER METRICS

Different kinds of metrics are used by management to measure the growth or change. For example, to measure the project progress, earned value analysis is used. From customer perspective there are different metrics like User satisfaction index, volume of repeated business be a customer, business obtained through
referrals, revenue savings and others. Organizations use their internal metric and industry standards to monitor the progress and maintain quality of software products.

III. PROCESS MODELS VS SOFTWARE QUALITY

Process models were evolved with the growth of software and demands of customers. Until 1980s, waterfall model was the prominent model used for software development. Later feedback loops were added to it, representing a step closure to improve quality [6]. SDLC added ETVX (Entry-Task-Verification-Exit) as a measure to improve the software quality. To meet customer needs, software organizations have come up with a prototype model to show case feasibility and look & feel of the final product and buy-in customer confidence. Thus resulting reduced rejections at the cost of increased scrap and time delays. The prototype is iterative and customer centric model.

Later spiral model [8] project type (software process model) showed a paradigm shift in the software quality. The approach advocates prevention by taking well defined scope and completing the task and later taking the next set of functions to be developed on the just developed product. This model reduced uncertainty resulting better quality product.

In V Model, testing is suggested in concurrent to the phases of the SDLC, thus defining the metrics for each phase and improving quality.

Unified Process Model revolutionized the thinking of architects and defined multiple measurable metrics and is still evolving. This model attempts to draw best features and characteristics of conventional software models. The Object Orient process resulted as a brainchild of Unified Process Model with defined metrics measured.

Component Based Development Model defined metrics related to component cohesion and coupling. Agile Modeling [9] is a practice-based methodology for effective modeling and documentation of software-based systems. This model was developed to facilitate the rapid development of operational software. This is the customer and industry driven model currently. This lead to explore new measurable metrics which changed the face of software industry and quality.

The impact of process models on various factors is depicted in the Figure 1 [10]. As the figure indicates, Quality of Customer, Quality of Design factors are increased and the Delivery Time and Bureaucracy factors is decreased in Agile Model.

These process models are mostly organizations/customer driven and these shown some improvement in quality, if not significant.

Software Quality can be viewed in five perspectives [11]. These are

- Transcendental View
- User View
IV. OBJECT ORIENTED AND AGILE MODELS

Object oriented methodology and agile methodology are the evolutions of 21st century. Object oriented methods and analysis gained widespread software engineering community in early 1990s. These two have changed the face of design and implementation.

A. OBJECT ORIENTED METRICS

Class is the fundamental unit of an object oriented systems. The OO metrics are defined at design, analysis and operational level to indicate the quantitative and qualitative measures for OO systems.

Metrics are defined to measure the characteristics of object models. MOOD Metrics suite [12] is used to measure the inheritance mechanism using the Method Inheritance Factor (MIF) and Attribute Inheritance Factor (AIF) metrics. The suite also defines the coupling between the classes by the Coupling Factor (CF). As CF increases complexity of the system increases and invariably the maintainability will suffer. Polymorphism Factor (PF) metric from measures the polymorphic behaviour of classes taken together.

The general metrics [13] is include Number of scenario scripts, number of key classes, number of support classes, number of subsystems and Average number of support classes per key class.

Various metrics are proposed to measure the properties of the OO systems. Few more were proposed to check the complexity and maintainability of the applications. Adaptability, robustness are the key quality features of maintenance projects.

B. AGILE METRICS

Modern software development is driven by the need to be agile. Agile was first introduced in 2001, by Agile Alliance [9]. This alliance defined 12 principles to follow the agile methodology. The overall agile framework is around the iterative and incremental processes and the Figure 2 depicts the same [14].

Agility implies dynamism, context based changes and growth. This model is another step bringing designer’s quality closure to customer’s quality view.

![Figure 2 – The Agile Framework](image-url)
A number of approaches are defined to quantify agility. Agility Index Measurements measures on five dimensions (duration, risk, novelty, effort and interaction). Another study using fuzzy mathematics suggests that project velocity can be used a metric for agility.

Below are few metrics measures used by project management groups to check the progress of projects.

- Sprint Goal success rate
- Defects
- Total Project Duration
- Time to Market
- Total project cost
- Team members turnover

Most organizations are targeting to understand the key factors and derive new metrics to suit their needs while their development is traditional centric.

V. INDUSTRIAL QUALITY STANDARDS

In view of the revolutionary changes in software, process should also scale up to suite the type and size of the project. Different quality and industrial standards notably CMMI, PMBOK Guide, ITIL and PRINCE2 recommend different guidelines and standards to enable achieving desired outcome from projects. Software organizations use process improvement to achieve their goals. One of the objectives is to improve the quality of the product. This can be achieved by reducing errors, improving good working environment, adopting best practices and following industrial quality standards.

VI. CONCLUSION

Metrics are key to measure, without measuring, we cannot complete projects successfully and measure the quality of the deliverable. Metrics are generated by collecting and assimilating related measures over a period of time across similar processes or applications. Software metrics evolved with the changing nature of software. Metrics are used as yardstick to measure progress and quality of the products developed. This paper analyse various process flows, the conventional, and evolutionary and object models and also how metrics changes as per industry needs. As in 1970s and 1980s, the practitioners developed measures to suit the needs and were able to show successful outcomes. Objected oriented metrics were introduced in the recent past to assist the development and monitor the SDLC of a software product. Practitioners are still using conventional metrics for Agile Methodology processes. Extreme Programming (XP) Development and Scrum Development follow the Agile Methodologies. New metrics are evolving to suit Agile Model, but the practitioners are still using the conventional metrics in measuring the progress and quality, thus resulting a gap to fill.

REFERENCES

15. Mr. S. Manivannan, Dr. S. Balasubramanian, “Software Metric Analysis Methods For Product Development / Maintenance Projects” International journal of Computer Engineering & Technology (IJCET), Volume 1, Issue 1, 2010, pp. 18 - 33, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.