ADVANCED SOFTWARE ENGINEERING

CSE 805

MSC

BY

Dr. Chinwe
Peace Igiri

Mountain Top University

SOFTWARE METRICS

CONCEPT

DEFINITION

Quantitative measure of degree to which a system, component or process possess a given attribute (IEEE)

Directly or indirectly observable quantities or numerical related to software development project management activities

Portability, Re-usability, Correctness, Reliability, Efficiency, Usability, Integrity, Maintainability and Flexibility.

CATEGORIES

Product metrics

Product metrics describe the characteristics of the product such as size, complexity, design features, performance, and quality level.

Process metrics

Process metrics can be used to improve software development and maintenance. Examples include the effectiveness of defect removal during development, the pattern of testing defect arrival, and the response time of the fix process.

Project metrics

Project metrics describe the project characteristics and execution. Examples include the number of software developers, the staffing pattern over the life cycle of the software, cost, schedule, and productivity.

Portability

Portability, in relation to software, is a measure of how easily an application can be transferred from one computer environment to another. A computer software application is considered portable to a new environment if the effort required to adapt it to the new environment is within reasonable limits. The meaning of the abstract term 'reasonable' depends upon the nature of the application and is often difficult to express in quantifiable units.

The phrase "to port" means to modify software and make it adaptable to work on a different computer system. For example, to port an application to Linux means to modify the program so that it can be run in a Linux environment..

RE-USUABILITY

Is the use of exiting software asset to develop a new one in order to reduce time and cost

A good software reuse process facilitates the increase of productivity, quality, reliability, and the decrease of costs and implementation time.

Has a positive impact on the quality and maintainability of software products.

LEVELS OF RE-USUABILTY

Code reuse, design reuse, specification reuse and application system reuse

CORRECTNESS

The adherence to the specifications that determine how users can interact with the software and how the software should behave when it is used correctly.

Rules

- Defining the problem completely.
- Develop the algorithm and then the program logic.
- Reuse the proved models as much as possible.
- Prove the correctness of algorithms during the design phase.
- Developers should pay attention to the clarity and simplicity of your program.
- Verifying each part of a program as soon as it is developed.

RELIABILITY

- It is the **probability of failure-free operation of a computer program for a specified period in a specified environment**.
- Reliability is a customer-oriented view of software quality.
- It relates to operation rather than design of the program, and hence it is dynamic rather than static.

- Software Reliability is an essential connect of software quality, composed with functionality, usability, performance, serviceability, capability, installability, maintainability, and documentation.
- Software Reliability is hard to achieve because the complexity of software turn to be high.
- While any system with a high degree of complexity, containing software, will be hard to reach a certain level of reliability, system developers tend to push complexity into the software layer, with the speedy growth of system size and ease of doing so by upgrading the software.

RELIABILITY METRIC

- Reliability metrics are used to quantitatively expressed the reliability of the software product. The option of which metric is to be used depends upon the type of system to which it applies & the requirements of the application domain.

Mean Time to Failure (MTTF)

MTTF is described as the time interval between the two successive failures. An **MTTF** of 200 mean that one failure can be expected each 200-time units. The time units are entirely dependent on the system & it can even be stated in the number of transactions. **MTTF** is consistent for systems with large transactions.
For example, It is suitable for computer-aided design systems where a designer will work on a design for several hours as well as for Word-processor systems.

To measure **MTTF**, we can evidence the failure data for n failures. Let the failures appear at the time instants $t_1, t_2 \ldots\ldots t_n$.
MTTF can be calclated as

$$\sum_{i=1}^{n} \frac{t_{i+1} - t_i}{(n-1)}$$

RELIABILITY METRIC

Mean Time to Repair (MTTR)

Once failure occurs, some-time is required to fix the error. **MTTR** measures the average time it takes to track the errors causing the failure and to fix them.

Mean Time Between Failure (MTBR)

We can merge **MTTF** & **MTTR** metrics to get the MTBF metric.

MTBF = MTTF + MTTR

Thus, an **MTBF** of 300 denoted that once the failure appears, the next failure is expected to appear only after 300 hours. In this method, the time measurements are real-time & not the execution time as in **MTTF**.

Rate of occurrence of failure (ROCOF)

It is the number of failures appearing in a unit time interval. The number of unexpected events over a specific time of operation. **ROCOF** is the frequency of occurrence with which unexpected role is likely to appear. A **ROCOF** of 0.02 mean that two failures are likely to occur in each 100 operational time unit steps. It is also called the failure intensity metric.

RELIABILITY METRIC

Probability of Failure on Demand (POFOD)

- **POFOD** is described as the probability that the system will fail when a service is requested. It is the number of system deficiency given several systems inputs.
- **POFOD** is the possibility that the system will fail when a service request is made.
- A **POFOD** of 0.1 means that one out of ten service requests may fail.**POFOD** is an essential measure for safety-critical systems. POFOD is relevant for protection systems where services are demanded occasionally.

Availability (AVAIL)

- Availability is the probability that the system is applicable for use at a given time. It takes into account the repair time & the restart time for the system.
- An availability of 0.995 means that in every 1000 time units, the system is feasible to be available for **995** of these.
- The percentage of time that a system is applicable for use, taking into account planned and unplanned downtime. If a system is down an average of four hours out of 100 hours of operation, its **AVAIL** is 96%.

EFFICIENCY

- Efficiency, in software development, is simply the amount of software developed or requirement meant divided by the amount of resources used like time, effort, etc. Different teams and different engineers will have different efficiency rates.

- Efficiency is generally more when the same task is done by the same team over and over again.

1. Meeting Times
2. Lead Time
3. Code Churn
4. MTTR and MTBF
5. Impact

EFFICIENCY METRICS

Meeting Times

- Meeting times can surprisingly predict a lot about the efficiency of a team of software developers.
- If the team is extending their meeting times on a regular basis, then the team is going through a problem.
- At the end of every sprint session, team leaders or scrum masters should inspect the committed meeting time and the actual meeting time.

Lead time

- Lead time is the amount of time between the birth to the end of a process.
- It starts right back from the initial discussion or proposal of the idea or requirement to the development and delivery of it.
- It depends on both team quality and project complexity, both of which directly affect the project cost

EFFICIENCY METRICS (2)

Code Churn

- Code Churn is the time the developers spend editing, adding, or deleting their own code.
- Basically, it measures the rate at which the code is evolving.
- Undoubtedly, code churn varies with project types, team quality and the course of the software life cycle.

MTTR and MTBF

- MTTR stands for Mean Time to Repair and MTBF means Mean Time to Failure.
- After successful software development and delivery, it is essential to fix issues like bugs and make efficient changes.
- These factors//metrics help in keeping track of loose ends even after software/requirement delivery.
- MTTR is the average time required to fix a failed or repair a bug or issue. Similarly, MTBF is the records the failures due to design conditions.
- **MTTR = (total maintenance time)/(total number of repairs)**
- **MTBF = (total operational time)/(total number of failures)**
- These metrics are useful because taking too long to repair can prove to be highly unpleasant and can impact the business.

EFFICIENCY METRICS(3)

Impact

- Impact is a measure of how the code is affected by the changes made in the code.
- It’s quite self-explanatory.
- The impact of a change set depends on a variety of factors such as amount of code, the seriousness of the code and the number of dependent functions and also files.
- A change in code that affect multiple files with addition and deletion at multiple locations will have more impact than a change made by adding a hundred lines of code in one file.

USABILITY

CONCEPT

- Usability refers to the quality of a user's experience when interacting with products or systems, including websites, software, devices, or applications.
- Usability is about effectiveness, efficiency and the overall satisfaction of the user.
- Usability Evaluation focuses on how well users can learn and use a product to achieve their goals.
- It also refers to how satisfied users are with that process.
- To gather this information, practitioners use a variety of methods that gather feedback from users about an existing site or plans related to a new site.

IMPROVE USABILITY

- Baseline usability testing on an existing site
- Focus groups surveys or interviews to establish user goals
- Card Sort testing to assist with IA development
- Wireframe testing to evaluate navigation
- First click testing to make sure your users go down the right path
- Usability testing to gauge the user interaction end-to-end and
- Satisfaction surveys to see how the site fares in the real world.