Software Quality Metrics

By,

Swathi Bhat

ICIS

- Software quality metrics are quantitative measures (measuring a quantity) used to assess (evaluate) the quality of software products or processes.
- focuses on collecting information that is not numerical. (Quality is not something that you measure with numbers.)
- Software quality and other software engineers have formulated the main objectives for software quality metrics

Frame 21.1 Main objectives of software quality metrics

(1) **To facilitate management control as well as planning and execution of the appropriate managerial interventions.** Achievement of this objective is based on calculation of metrics regarding:

- Deviations of actual functional (quality) performance from planned performance
- Deviations of actual timetable and budget performance from planned performance.

(2) **To identify situations that require or enable development or maintenance process improvement in the form of preventive or corrective actions introduced throughout the organization.** Achievement of this objective is based on:

- Accumulation of metrics information regarding the performance of teams, units, etc.

- Managerial interventions refer to actions taken by managers within an organization to address (fix) various issues, improve performance, resolve conflicts.
- Accumulation of metrics information regarding the performance of teams, units, etc. refers to the process of collecting, gathering, and storing various data points and measurements related to the performance of these entities within an organization.

Classification of software quality metrics

- Software quality metrics can fall into a number of categories. Here we use a two-level system.
- The first classification category distinguishes between life cycle and other phases of the software system:
- ■ Process metrics, related to the software development process
- ■ Product metrics, related to software maintenance

- The second classification category refers to the subjects of the measurements:
- ■ Quality
- ■ Timetable
- ■ Effectiveness (of error removal and maintenance services)
- ■ Productivity.

- A sizeable number of software quality metrics involve one of the two following measures for system size (In the context of software quality metrics, the term "system size" typically refers to the measure of the complexity of a software system. There are two common measures used to quantify system size):
- **KLOC** (Thousand Lines of Code) and **Function Points** are both metrics used in software development to measure the size and complexity of software projects.

- **KLOC (Thousand Lines of Code)**:
- KLOC simply measures the size of a software project by counting the lines of code.
- It's a straightforward metric used to estimate the effort required for development, maintenance, and testing of software.
- KLOC can be useful for estimating project costs and schedules, as well as for benchmarking productivity.

- **Function Points**:
- Function Points, on the other hand, measure the functionality provided by a software application based on the user's perspective.
- Function Points take into account the complexity and functionality of the software, including inputs, outputs, interfaces, and files.
- Function Points provide a more abstract measure compared to KLOC, focusing on what the software does rather than how it's implemented.
- Function Points are often used for estimating project effort, determining productivity, and comparing the size and complexity of different software systems.

Process metrics

- Process metrics in software development refer to the quantifiable measurements used to assess various aspects of the software development process.
- These metrics help teams understand the efficiency, effectiveness, and quality of their processes, enabling them to identify areas for improvement.
- Software development process metrics can fall into one of the following categories:
- ■ Software process quality metrics
- ■ Software process timetable metrics
- ■ Error removal effectiveness metrics
- ■ Software process productivity metrics.

Software Process Quality Metrics

- These metrics evaluate the quality of the software development process itself.
- This includes factors such as supporting to coding standards, consistency in documentation, thoroughness of testing procedures, and the effectiveness of quality assurance measures.
- By analyzing these metrics, teams can identify areas where the process may be lacking and implement improvements to enhance overall software quality and reliability.

- Software process quality metrics may be classified into two classes:
- ■ Error density metrics
- ■ Error severity metrics.

- **Error Density Metrics**:
- These metrics focus on quantifying the frequency or density of errors within the software.
- They typically measure aspects such as the number of defects per unit of code (e.g., lines of code, function points, or modules), the number of defects discovered during specific phases of development (e.g., requirements, design, coding, testing), or the defect density in different modules or components of the software (UI/DB).
- The goal of error density metrics is to identify areas of the software that are more error-prone and may require additional attention or improvement.

- **Error Severity Metrics**:
- Error severity metrics, on the other hand, evaluate the impact or seriousness of errors found in the software.
- Instead of just counting the number of defects, these metrics categorize defects based on their severity levels, such as critical, major, minor.
- Severity levels are typically defined based on the impact of the defect on the functionality, usability, performance, security, or other quality attributes of the software.
- By analyzing the distribution of defects across different severity levels, teams can prioritize their efforts to address critical issues first and allocate resources more effectively to improve overall software quality.

Software Process Timetable Metrics:

- Timetable metrics focus on the time-related aspects of the software development process.
- This includes metrics like time to market, cycle time, lead time, and related to project deadlines.
- (This metric measures the duration it takes for a product or service to be developed, tested, and launched to the market.
- Cycle time represents the total time taken to complete one cycle of a process, from the initiation to the delivery of a product or service.
- Lead time is the duration between the initiation and the completion of a process. It includes all the time spent on actual work as well as any waiting time or delays in between.)
- Timetable metrics are crucial for evaluating the efficiency of the development process, identifying bottlenecks, and optimizing workflow to ensure timely delivery of software products or updates.

Error Removal Effectiveness Metrics:

- These metrics measure the efficiency and effectiveness of error detection and removal processes throughout software development.
- They encompass metrics such as defect density, defect removal efficiency, mean time to detect (MTTD), and mean time to repair (MTTR).
- By tracking these metrics, teams can evaluate the efficacy of their quality control measures and make adjustments to minimize the occurrence of errors in the final product.

Software Process Productivity Metrics:

- Productivity metrics measure the efficiency of the software development process in terms of resource utilization and output.
- This includes metrics such as lines of code written per hour, number of features implemented within a given timeframe.
- These metrics help teams evaluate their productivity levels, identify areas for improvement, and make informed decisions to optimize resource allocation and streamline development workflows.
- (Streamlining development workflows refers to the process of optimizing and improving the efficiency of various tasks and procedures involved in software development.)

Product metrics

- Product metrics refer to the software system's operational phase (maintenance phase) – years of regular use of the software system by customers, whether "internal" or "external" customers, who either purchased the software system or contracted for its development.
- In most cases, the software developer is required to provide customer service during the software's operational phase. Customer services are of two main types:

- **Help desk services (HD)** – software support by instructing customers regarding the method of application of the software and solution of customer implementation problems.
- Demand for these services depends to a great extent on the quality of the user interface (its "user friendliness") as well as the quality of the user manual and integrated help menus.
- (demand for certain services relies heavily on how easy their user interface is to use,)

- **Corrective maintenance services** – correction of software failures identified by customers/users or detected by the customer service team prior to their discovery by customers.
- The number of software failures and their density are directly related to software development quality.
- For completeness of information and better control of failure correction, it is recommended that all software failures detected by the customer service team be recorded as corrective maintenance calls.

- Commonly, all customer services – namely, HD and corrective maintenance services – are provided to customers/users by a software support centre (the "customer service centre").
- It is expected that very few customer calls will be related to identified failures.

- The array of software product metrics presented here is classified as follows:
- ■ HD quality metrics
- ■ HD productivity and effectiveness metrics
- ■ Corrective maintenance quality metrics
- ■ Corrective maintenance productivity and effectiveness metrics.

- **HD quality metrics**
- The types of HD quality metrics discussed here deal with:
- ■ HD calls density metrics – the extent of customer requests for HD services as measured by the number of calls.
- ■ Metrics of the severity of the HD issues raised.
- ■ HD success metrics – the level of success in responding to these calls.
- A success is achieved by completing the required service within the time determined in the service contract.

- **HD productivity and effectiveness metrics**
  Productivity metrics relate to the total of resources invested during a specified period, while effectiveness metrics relate to the resources invested in responding to a HD customer call.

- **Corrective maintenance quality metrics**
- Software corrective maintenance metrics deal with several aspects of the quality of maintenance services.
- A distinction is needed between software system failures treated by the maintenance teams and failures of the maintenance service that refer to cases where the maintenance failed to provide a repair that meets the designated standards or contract requirements.
- Thus, software maintenance metrics are classified as follows:

- **Software system failures density metrics** – deal with the extent of demand for corrective maintenance, based on the records of failures identified during regular operation of the software system.
- ■ **Software system failures severity metrics** – deal with the severity of software system failures attended to by the corrective maintenance team.

- **Failures of maintenance services metrics –** deal with cases where maintenance services were unable to complete the failure correction on time or that the correction performed failed.
- ■ **Software system availability metrics–** deal with the extent of disturbances caused to the customer as realized by periods of time where the services of the software system are unavailable or only partly available.

- **Software corrective maintenance productivity and effectiveness metrics**
- While corrective maintenance productivity relates to the total of human resources invested in maintaining a given software system, corrective maintenance effectiveness relates to the resources invested in correction of a single failure.
- a more effective software maintenance system will require fewer resources, on average, for correcting one failure.

- Thank You