Assignment 2

Student Name: Sahil Kaundal UID: 21BCS8197

Branch: CSE (Lateral Entry) Section/Group:20BCS-807\_B Semester: 4th Date of Performance: 11/04/2022

Subject Name: SE Subject Code: 20CSP-254

Q1.

Explain software quality metrics for process and products. How these metrics will be helpful for engineers?

Answer:

Software quality metrics are a subset of software metrics that focus on the quality aspects of the product, process, and project. These are more closely associated with process and product metrics than

Product Quality Metrics

This metric includes the following −

* Mean Time to Failure: This metric is mostly used with safety critical systems such as the airline traffic control systems, avionics, and weapons.

* Defect Density: It measures code quality per unit.

* Customer Problems: It measures the problems that customers encounter when using the product.

* Customer Satisfaction: Customer satisfaction is often measured by customer survey data through the five-point scale −Very satisfied, Satisfied, Neutral, Dissatisfied, Very dissatisfied

In-process Quality Metrics:

In-process quality metrics deals with the tracking of defect arrival during formal machine testing for some organizations. This metric includes −

* Defect density during machine testing
* Defect arrival pattern during machine testing
* Phase-based defect removal pattern
* Defect removal effectiveness

Benefits:

Engineering metrics provide valuable benefits for organizations that develop software.

* They can help increase the overall quality of the product.
* They can increase team productivity.
* Metrics can even help you reduce costs by alerting you early on about problems that may show up in all stages of the software development life cycle.

Q2.

Elaborate on the fact that architecture is the vehicle for stakeholder communication.

Answer:

Architecture is the vehicle for stakeholder communication:

Each stakeholder of a software system-customer, user, project manager, coder, tester, and so on-is concerned with different system characteristics that are affected by the architecture. For example, the user is concerned that the system is reliable and available when needed; the customer is concerned that the architecture can be implemented on schedule and to budget; the manager is worried (as well as about cost and schedule) that the architecture will allow teams to work largely independently, interacting in disciplined and controlled ways. The architect is worried about strategies to achieve all of those goals.

Architecture provides a common language in which different concerns can be expressed, negotiated, and resolved at a level that is intellectually manageable even for large, complex systems. Without such a language, it is difficult to understand large systems sufficiently to make the early decisions that influence both quality and usefulness.

The Architecture Defines Constraints on Implementation An implementation exhibits an architecture if it conforms to the structural design decisions described by the architecture. This means that the implementation must be divided into the prescribed elements, the elements must interact with each other in the prescribed fashion, and each element must fulfil its responsibility to the others as dictated by the architecture. Resource allocation decisions also constrain implementations. These decisions may be invisible to implementors working on individual elements. The constraints permit a separation of concerns that allows management decisions to make the best use of personnel and computational capacity. Element builders must be fluent in the specification of their individual elements but not in architectural trade-offs. Conversely, architects need not be experts in all aspects of algorithm design or the intricacies of the programming language, but they are the ones responsible for the architectural trade-offs.

Communication architecture defines the frequency and fidelity of information flow between individuals in your organization. It helps structure how and when you communicate, both within a team and cross-functionally. Having a well-understood communication architecture can help you diagnose trouble spots, so you’re always iterating and improving.

For example, does your engineering team understand your customers’ problems so they can build the right solution? If not, perhaps there’s a breakdown in communication between your sales and engineering teams. Perhaps a monthly meeting between the teams, an email, or some other medium will help. Three months in, does a new hire understand the company’s mission and where it’s going? If not, perhaps your onboarding process could use some work. The specific tactics are unique to each organization, but it requires proactive thought and investment.

Creating a communication architecture starts with understanding your company’s current goals and people. Are you an early-stage startup in a rapidly changing environment with competitors breathing down your neck Or do you already have a solid foundation, product, and team, and are now focusing on adding structure and scaling.