Software Quality Engineering

Assignment - 1

Kiran Chauhan
Roll No -> 28
Sec -> A

Q-> (1) Why is defect prevention, defect Reduction and Defect containment important in Software Quality Engineering?

Ans -> Defect Prevention, Defect Reduction and Defect Containment are necessary procedure for delivering a quality & defect free software product.

Defect prevention has a great impact on improving the software process by backing up the testing process and reducing the cost of fixing errors.

Fixing errors is cheaper when they are detected early in the process.

Defect prevention activities are also a mechanism for communicating lessons learned among software projects and groups.

Rework has a considerable impact on production cost and It has been the main reason for delays in the development process.

Defect Prevention reduces the amount of rework and ensures low production cost and fast delivery.

Defect containment activities helps in maintaining the negative impact of remaining faults during operational use after product release.

Defect Prevention activities are intended to improve quality and improve productivity.

Q→ (2) In your project, do you have a defend defect handling process? If so, describe the process and compare it with the generic description. If not design one to specify defect handling process.

Ans → No, there is no defect handling process in my project. But to handle defect, I will design a defect handling process like defect prevention.

Defect prevention is usually considered as a process of simply preventing defects from their resources in software development life cycle.

It is basically defined as a measure to ensure that defects being detected so far, should not appear or occur again.

Defect prevention is the best method to eliminate the defects in the later stage and their fixing it.

However, it is not possible to remove all defects but at least you can minimize the impact of the defect and cost to fix the same.

The major steps involved in defect prevention are as follows :-

(1) Identify Critical Risk
(2) Eliminate Expected impact
(3) Minimize expected impact

Kiran Chauhan
Roll No→28
Sec → A

Q→③ Explain the following software Quality Models :-

(a) ISO/IEC 9126 Model :-

The ISO/IEC 9126 standard describes a software quality model which categories software quality into six characteristics which are sub divided into sub characteristics :-

Six main quality factors are :-

① Functionality →
When the software is used under specified conditions, the software product's ability to provide functions that met explicit and implicit needs.

② Reliability →
The ability of a software product to maintain a specified performance level when used under specific conditions.

③ Usability → When used under specific conditions the software product's ability to understood, learned, used and attracted to users.

④ Efficiency → Under specified conditions, relative to the amount of resources used.

⑤ Maintainability → The ability of a software product to be modified. Modifications may include corrections, improvements or the degree of adaption of the software to changes in the environment, requirements and functional specifications.

⑥ Portability → The ability of a software product to migrate from one environment to another.

Kiran Chauhan
Roll No → 28
Sec → A

(b) ISO/IEC 25010 : 2011 :

A product quality model composed of eight characteristics which are further subdivided into sub characteristics that relate to the static properties of software.

(1) Functional Suitability → This characteristics represents the degree to which a product provides functions that meet stated and implied needs when used under specific conditions.

(i) functional completeness
(ii) functional correctness
(iii) functional appropriateness

(2) Performance Efficiency → This characteristics represents the performance relative to the amount of resources used under stated conditions.

(i) Time Behavior
(ii) Resource utilization
(iii) Capacity

(3) Compatibility → Degree to which a product, system or component can exchange information with other products, system or components or perform its required functions while sharing the same hardware or software environment.

(i) Co existence
(ii) Interoperability.

Kiran Chauhan
Roll No → 28
Sec → A

Kiran Chauhan
Scc → A

④ Usability → Degree to which a product can be used by specified users to achieve specific goals with effectiveness.
(i) Appropriateness recognizability
(ii) Learnability (iii) Operability (iv) Accessibility
(v) User interface aesthetics
(vi) User error protection.

⑤ Reliability → Degree to which a system performs specified functions under specified conditions for a specified period of time.
(i) Maturity (ii) Fault Tolerance
(iii) Availability (iv) Recoverability.

⑥ Security → Degree to which a product protects information and data so that persons or other products have the degree of data access appropriate to their types and levels of authorization.
(i) Confidentiality (ii) Integrity (iii) Non Repudiation
(iv) Accountability (v) Authenticity.

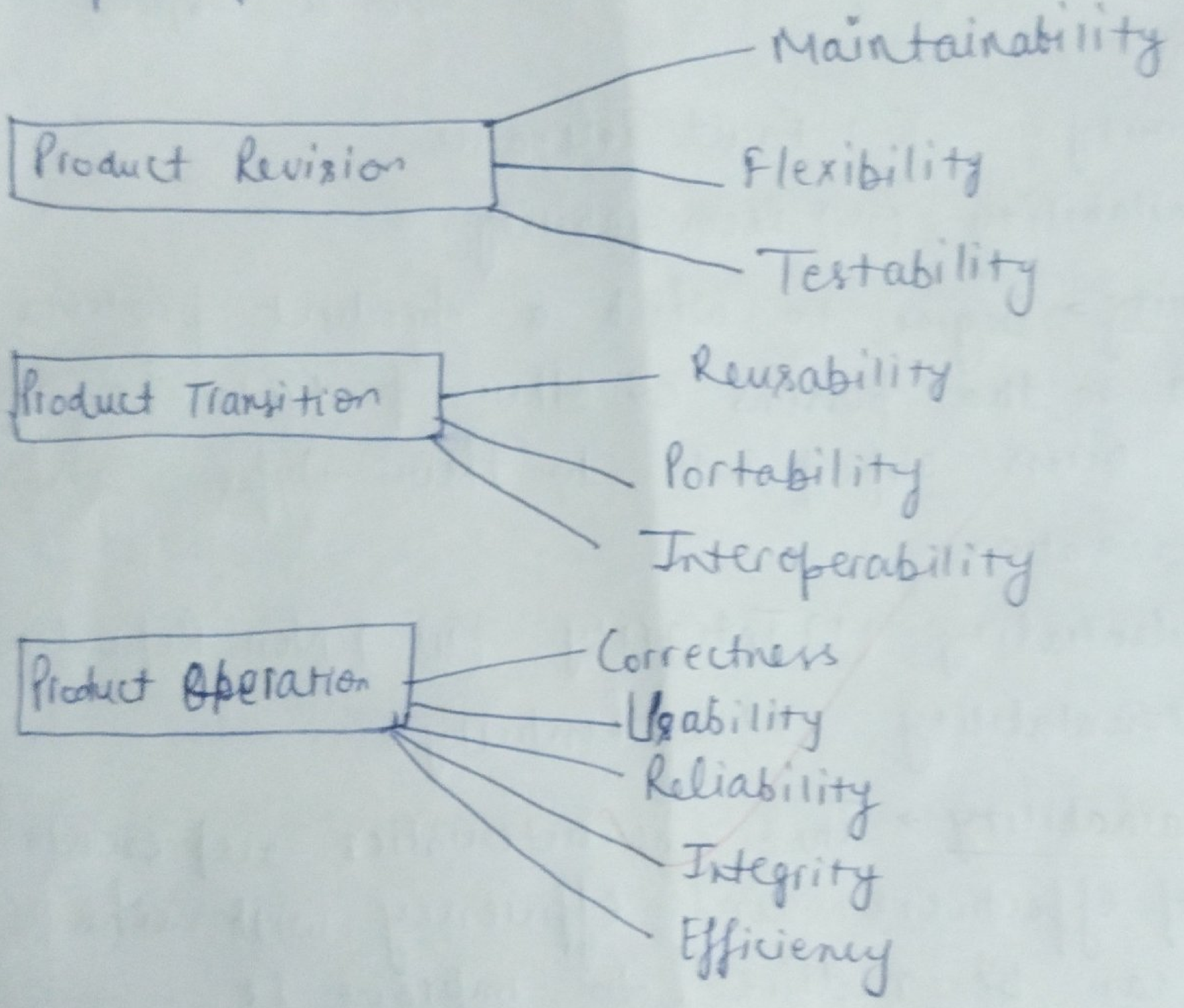
⑦ Maintainability → This characteristics represents the degree of effectiveness and efficiency with which a product can be modified to improve it.
(i) Modularity (ii) Reusability (iii) Analysability
(iv) Modifiability (v) Testability.

⑧ Portability → Degree of effectiveness and efficiency with which a system can be transferred from one hardware, software or other operational or usage environment to another.
(i) Adaptability (ii) Installability (iii) Replaceability

Kiran Chauhan
Sec → A

(c) McCall's Quality Model.

McCall's model was first introduced in the US Airforce in the year 1977
This quality model attempts to bridge the gap between users and developers by focusing on a number of software quality factor that reflect both
- User's view
- developer's priorities



(d) Boehm Quality Model

Boehm Quality model was introduced in the year 1978. It was a kind of hierarchial model that is structured around high level characteristics.
Boehm model measures software quality based on certain characteristics

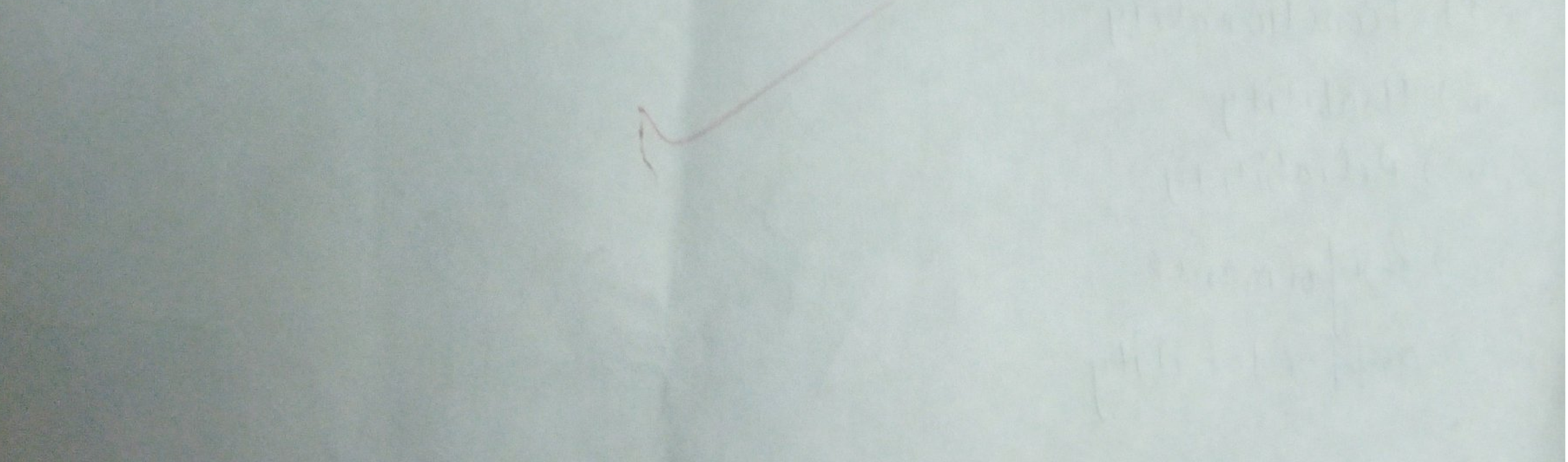
f) Dromey's Quality Model -

It is mainly focused on the attributes and sub attributes to connect properties of the software to quality attributes.

Dromey's software quality model is created by Dromey and presents four quality categories where each category consists of quality attributes

There are three principal elements to this model

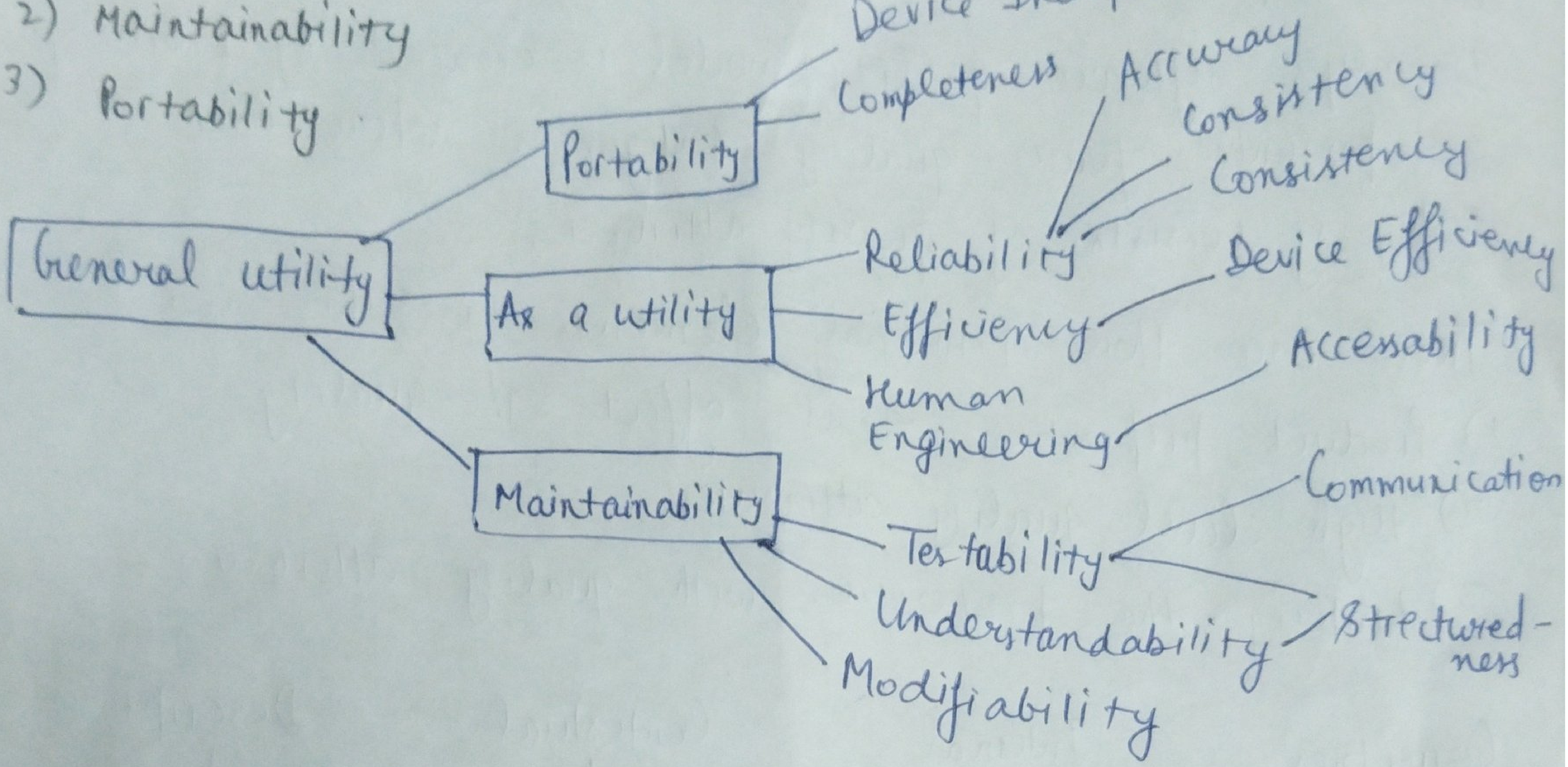
1) Product properties that affect the quality
2) High level quality attributes
3) Linking the properties with quality attributes.



Kiran Chauhan
Roll No → 28
Sec → A

The highest level of Boehm's model has following three primary uses :-

1) Act as utility
2) Maintainability
3) Portability



(e) FURPS Quality Model

FURPS is a quality model which was released on 1995. It is also known as quality characteristics table. The model was developed by Hewlett Packard was first publicly elaborated by Grady and Caswell.

(i) Functionality
(ii) Usability
(iii) Reliability
(iv) Performance
(v) Supportability

good

11.4.23

Kiran Chauhan
Roll No → 28
Sec → A