UNIT-V

- SOFTWARE RELIABILITY AND QUALITY MANAGEMENT:
  - SOFTWARE RELIABILITY
  - STATISTICAL TESTING
  - SOFTWARE QUALITY
  - SOFTWARE QUALITY MANAGEMENT SYSTEM
  - ISO 9000
  - SEI CAPABILITY MATURITY MODEL
- COMPUTER AIDED SOFTWARE ENGINEERING:
  - CASE AND ITS SCOPE
  - CASE ENVIRONMENT
  - CASE SUPPORT IN SOFTWARE LIFE CYCLE
  - OTHER CHARACTERISTICS OF CASE TOOLS
  - TOWARDS SECOND GENERATION CASE TOOL
  - ARCHITECTURE OF A CASE ENVIRONMENT.

• SOFTWARE RELIABILITY:

→ Reliability means within the time period the system (or) product doesn't failure.

→ Reliability is the one of the quality attribute.

→ The organization developed the quality products they are assurances to the customers.

→ The quality product has standard certificats these are ISO, CMM, Six sigma.

→ IEEE defines reliability as "software reliability is the probability of failure free operations of software over a given time interval and under given condition".

Reliability metrics :

MTBF (Mean Time Between Failure):

Mean Time Between Failure is measured by combining MTTR and MTTF.

MTBF = MTTR + MTTF.

If a failure is occured over a specified time duration, then it measures the occurrance of the next failure over the next duration. For example MTBF is 24 hours.

MTTR (Mean Time To Repair)

It is the average time taken to repair defects in a system either Hardware problem or Software problem. The system repair specified time intervals.

MTTF (Mean Time To Failure):

It is the average time interval between the consecutive failures observed over a large number of failures. It consider only run time.

Reliability approaches:

Fault tolerance : fault tolerance is the reliability approach, the main aim of fault tolerance is to overcome the faults, follow the series of activities these are

ⓐ Fault detection

ⓑ Fault assessment

ⓒ Fault removing (or) repairing.

• STATISTICAL TESTING:

The statistical testing focus on how faulty programs can affects its operation conditions.

Example: Customer uses ATM systems to withdrawal money Transfer money, change password, enquire balance while bank clerk performs transaction issues and security issues etc.

Example: ATM Operations find the statistical Testing.

Sol: The functions withdraw money through ATM machi may be broken down into two runs, one is withdraw money and another one is updating the balance in the account, Each type set of input variables.

performing statistical Testing:

→ Test cases are executed in intervals

→ failure rate of operation is calculated from the computed data.

• SOFTWARE QUALITY:

“Quality is depends on the quality attributes”, or quality factors.

The software quality factor can be two types.

1. Direct measured quality (functional)

2. Indirect measured quality (non functional)

Software quality factors: ① correctness ② Reliability ③ usability ④ Integrity ⑤ Efficiency ⑥ portability ⑦ Reusability ⑧ Interoperability ⑨ maintainability ⑩ Flexibility ⑪ Testability

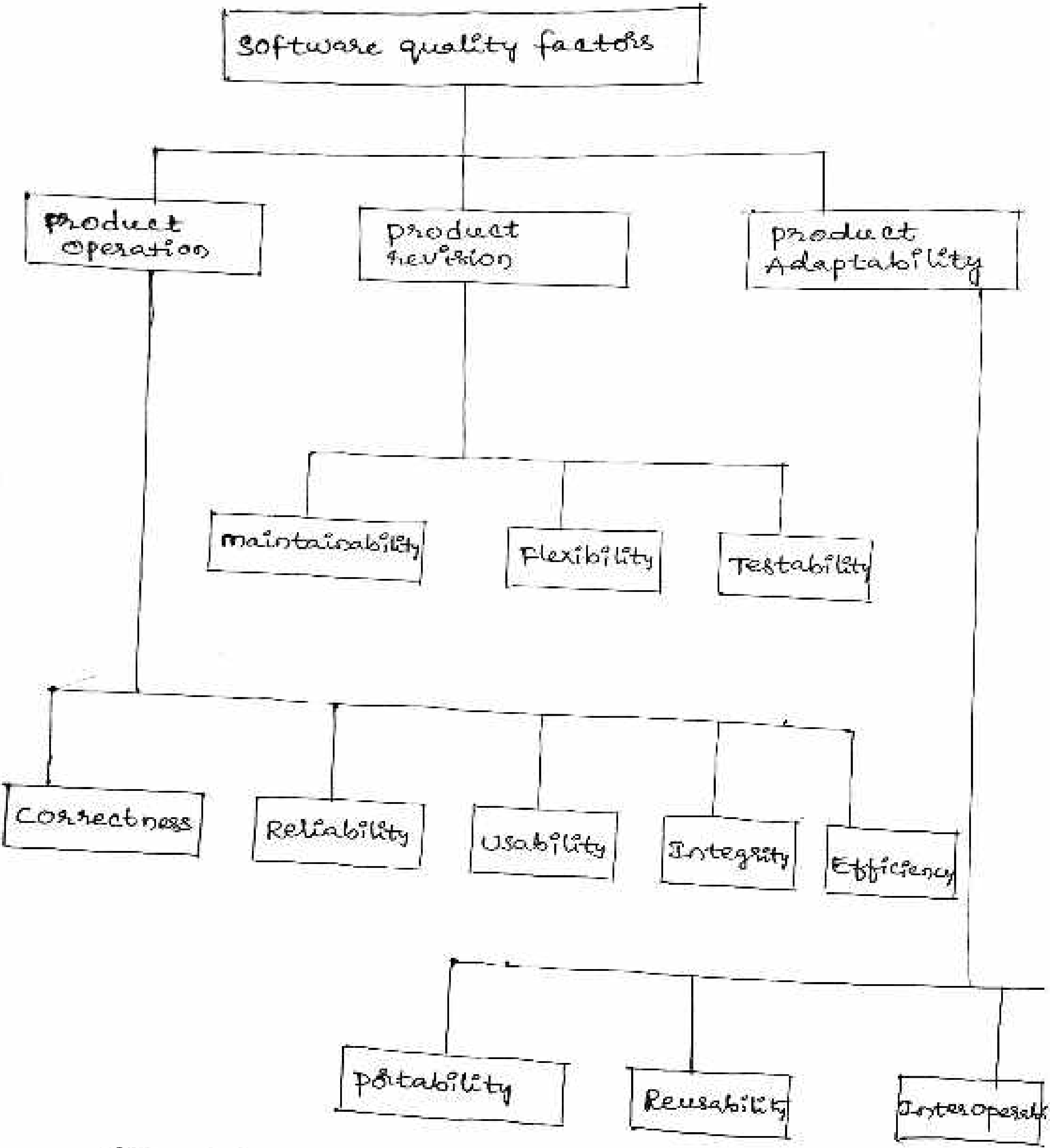

fig: Software quality factors.

• SOFTWARE QUALITY MANAGEMENT SYSTEM:

→ producing a high-quality product is the primary goal of an organization.

→ A quality management system (QMS) is a set of procedures/processes which are carried out to ensure that the product/services delivered by the organization have the desired quality.

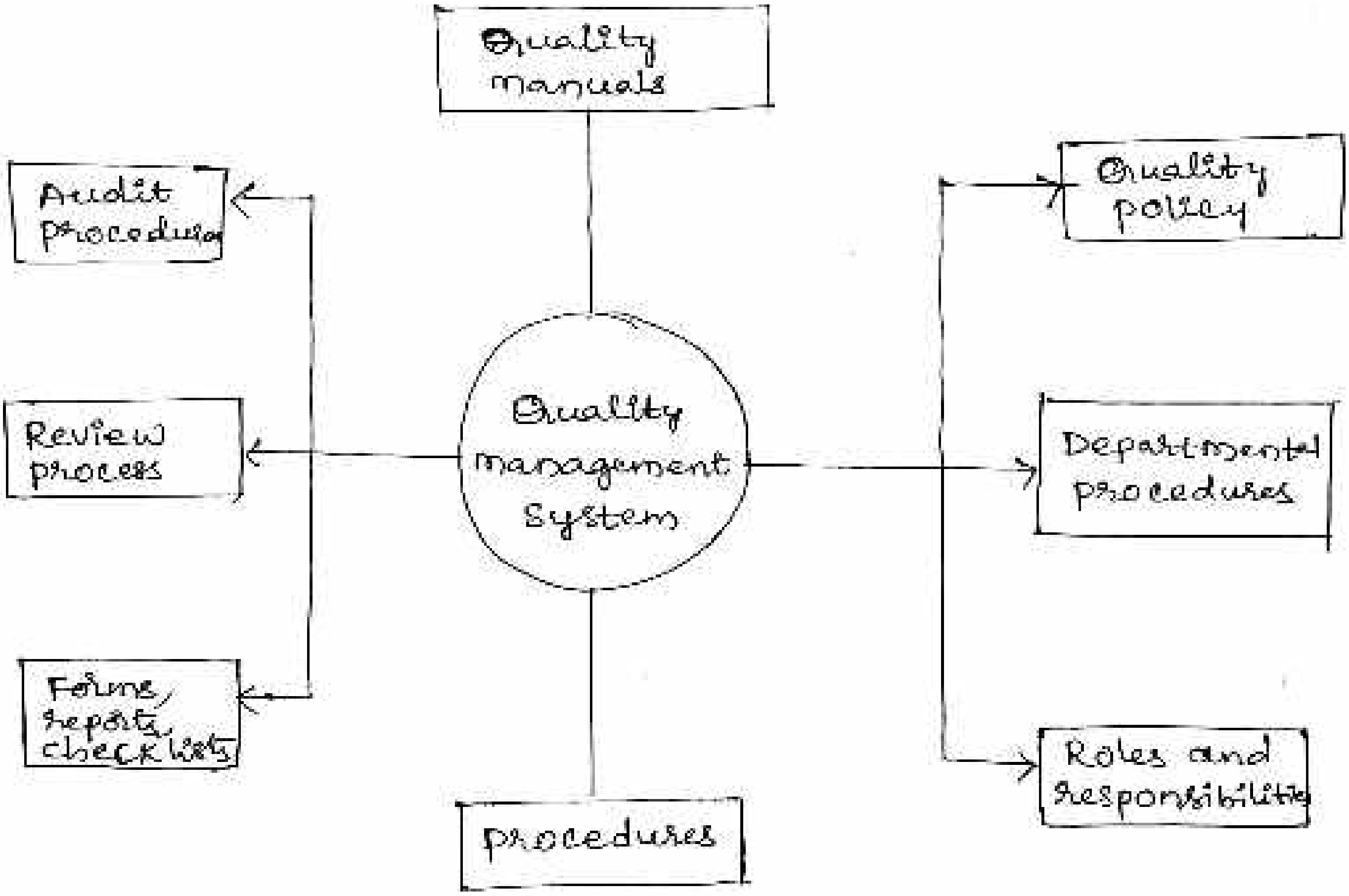

fig: Aspects of quality management system.

→ A QMS has well defined forms, reports and checklists based on the review process conducted by Audit team.

→ In the QMS, each individual has defined roles and responsibilities.

• ISO 9000 :

→ International standard organization (ISO) is a world wide fedaration.

→ It provide services, good practices, helping to make product in the industry more efficient and effective.

→ ISO may be Informational technologyical team and electrical technological team.

What is the ISO 9000 standard?

→ The ISO 9000 series is a set of international standards for quality management and quality assurance.

→ The ISO 9000 is applicable to various industries other than software industry.

→ ISO 9000 standards were published in 1987 as ISO 9000:1987, which were revised in 1994 as ISO 9000:1994.

→ The ISO 9000:1994 had three standards

(i) ISO 9001:1994
(ii) ISO 9002:1994
(iii) ISO 9003:1994

After that ISO 9002 :1994 and ISO 9003:1994 merged into ISO 9001:1994.

→ In the year 2000, ISO 9001:1994 was revised written in ISO 9001 : 2000

→ After revision ISO 9001 : 2008 is latest.

- ISO 9000 certification process:

→ Before going to get the ISO 9000 certificate, to know about the formal process of development.

→ An organization got ISO 9000 certificate, the people think that ~~th~~ the organizational products are developed qualitatevely.

→ How to get ISO 9000 certification apply to the registrar!

1. proposal stage
2. pre-assessment stage
3. Document review and adequacy audit
4. compliance audit
5. Registration
6. continued survelliance.

1. proposal stage:

The organization prepares a proposal in the format and according to the guidelines provided by the ISO to apply registrar for ISO certification.

2. pre-assessment stage:

At this stage registrar and the responsible people make a rough assessment of the organization.

3. Document review and adequacy audit.

The registrar reviews the proposal and its related documents if there is any flaw in the proposal. Audit confermations that organization has made the planned assessment of ISO certification.

4. Compliance audit: This is Verification process to check whether The organization has realized suggessions provided by registrar document review.

5. Registration: The organization receives the ISO certification from ISO registrar.

6. Continued surveillence: It is a temporary registration.

• SEI CAPABILITY MATURITY MODEL:

→ The capability maturity model (CMM) is an industry standard model for defining and measuring the maturity of the development process and for providing strategy for improving the software process toward achieving high quality products.

→ It was established by Software Engineering Institute (SEI) in 1986 at USA.

→ The main purpose of SEI is to help organizations improvement.

→ The CMM has five levels, these are

Level 1. Initial

Level 2. Repeatable

Level 3. Defined

Level 4. managed

Level 5. optimized.

Level 1. Initial:

The overall initiation of the project is the most important, it follows some standards and processes.

Level 2. Repeatable:

In repeatable level, software development successes are repeatable.

Level 3. Defined.

The organization has standardized processes for both management and development activities.

Level 4. managed,

Both the software process and products are quantitatively understood and controled.

Level 5. optimized,

To reduced the defects in the products, and improve the quality and productivity.

• COMPUTER AIDED SOFTWARE ENGINEERING (CASE):

→ Computer-aided software engineering (CASE) is a tools support for software development, project management, and maintanance activities.

→ These are the automated tools that help developers to perform software development and project management tasks.

→ CASE tools are well tested which assist to produce desired results and uncover flaws at any stage of development.

→ CASE tools reduces the development and maintanance costs.

→ CASE tools helps the developers, managers.

• CASE AND ITS SCOPE:

→ Software development is a complex process, which includes various activities, tools, methodology, and technology.

→ It helps to faster development and better product has been challenging issues for practitioners.

→ CASE is a tools support to automate the SDLC activities.

→ CASE tools are broadly Used in the following areas.

1. Requirement engineering
2. Software design
3. model generation (data model, DFD etc)
4. process description
5. User Interface design (UID).
6. Code generation
7. Test case generation
8. Document generation
9. Reverse engineering
10. Configuration management.
11. Version change control
12. project management
13. Screen and report generation
14. Project planning.

CASE tools are generally classified into three categories, these are

1. Upper CASE tools
2. Lower CASE tools
3. I-CASE tools (Integrated CASE tools)

Upper CASE tools:

This category of CASE tools support the development activities at the early stages of software development process. Such tools are diagramming tools, report and form generators and analysis tools.

Lower CASE tools:

These software tools deals with the implementation and integration tasks in software development. These tools support database schema generation, program generation, testing.

Integrated CASE tools:

The integrated CASE tools support, both, early and later stages of software development. The Upper CASE tools and lower CASE tools are integrated together with the help of CASE repository.

Advantages of CASE tools:

1. The CASE tools are used for routine development, management, planning and software maintenance activities.
2. These are helpful ~~a~~ in automating manual activities which reduces the effort of development and human mistakes.
3. It saves the development costs.
4. It improve the quality product.
5. Fast development process!

Disadvantages of CASE tools:

① Needs a lot of study and learning before using it.
② Training costs ③ Reduce manual testing in Industry.

• CASE ENVIRONMENT:

→ Software development life cycle activities can be assisted by the Use of software tools.

→ Programmers can use automated tools for development, maintenance, management and planning purposes.

→ The CASE environment of various automated tools is shown in figure.

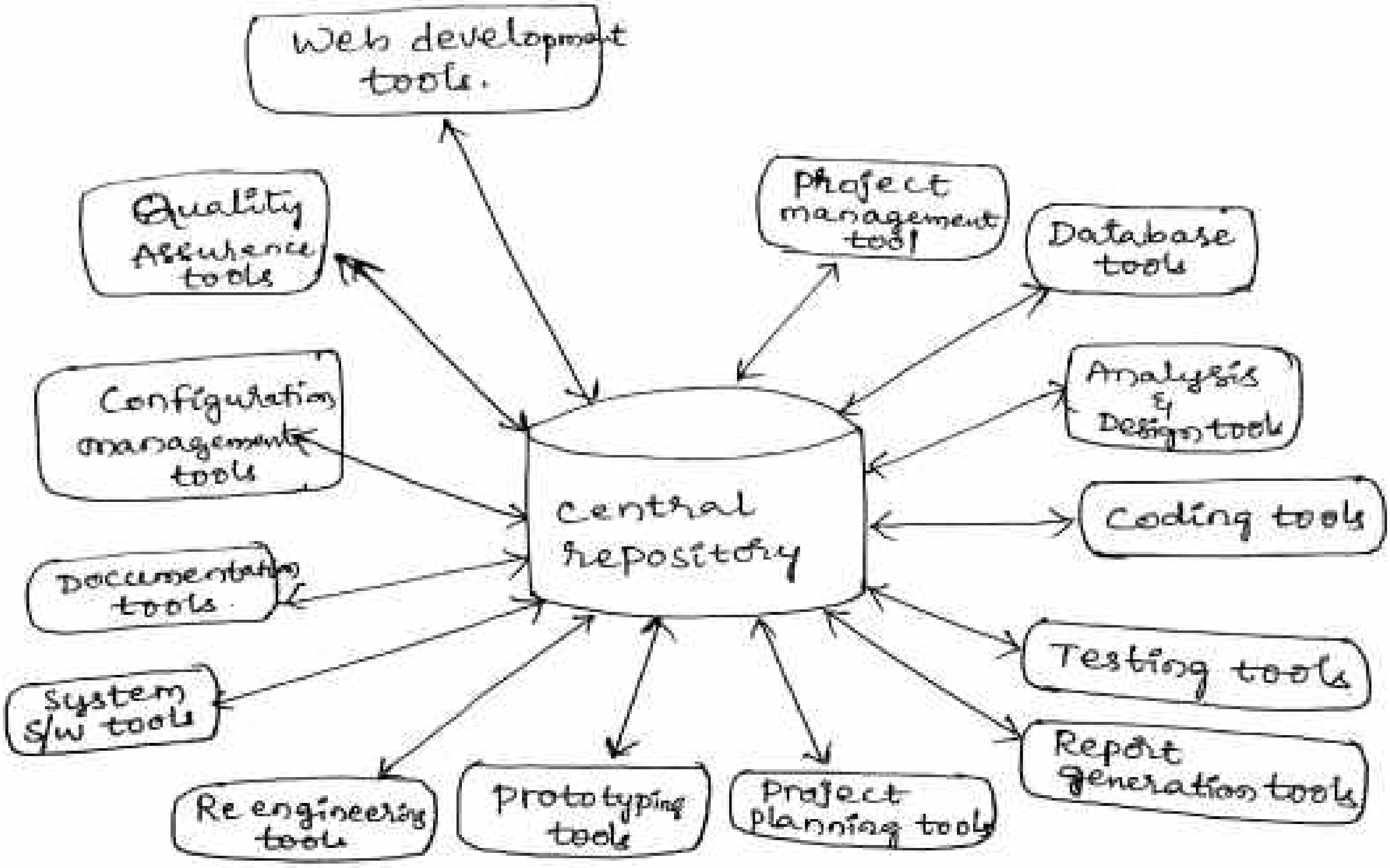

fig: CASE tools.

• Central repository :

It is a collection of stored data merged into one so that it may be shared, analysed, & updated throughout development. It holds diagrams, descriptions specifications, test data.

• Project management tools :

It can be used to design project schedule, activity plan etc.

Database management tools:

The database for relational and object oriented database management system is designed to provide support for CASE repository.

Analysis and design tools:

These tools help to produce analysis and design models of the system to be built.

programming tool:

compilers, editors, debugger, database query generator.

Integration and Testing tools:

These tools are basically used for data preparation static and dynamic measurement, simulation, test plan.

project planning tool:

These tools are useful for cost and effort estimation and project scheduling activities.

prototyping tools: These tools helps to design screen layouts, data design.

Reengineering tools:

~~I~~ These are used during system migration from an old platform to a new platform.
i.e online system reengineering tools.

System Software tools:

Various tools that assist to work in desktop and network systems. Ex: Network system software.

Documentation tools:

These tools are useful to produce documentation of the work products.

Software Configuration managent:

These activities are configuration identification, version control, change control.

Quality assurance tools: These are auditing tools used to determine compliance certificate process identification tools.

Report generation tools: create, modify, test prototype.

web development tools: Designing web applications, Applets, texts forms, Graphics.

• CASE SUPPORT IN SOFTWARE LIFE CYCLE :

→ CASE support in automating the various activities of software development process, project management, maintenance and reengineering activities.

→ CASE tools supports SDLC.

1. project planning and management – Effort estimation models, SLOC/FP
2. prototyping – Templates, GUI,
3. Analysis and design – DFD, ER modeling
4. Coding – EX: compiler, editors,
5. Testing – EX: Selenium, Jmeter
6. maintenance and reengineering – Automated updated tools.
7. Documentation – User manuels in diagrams, tools, tables etc.

• CHARACTERISTICS OF CASE TOOLS:

Some of the characteristics of CASE tools.

1. methodology
2. Flexibility
3. Integration
4. Validation
5. collaboration
6. Repository Interface
7. Testing software
8. Reverse engineering
9. Help system.

• TOWARDS SECOND GENERATION CASE TOOLS:

→ First generation CASE toolkit mainly includes the prototyping tools, graphical design tools, code generators

→ The graphical design tools help user to produce design diagrams based on selected methodology.

→ First generation CASE tools are lacking to bridge the gap between design and application generation.

→ Second generation CASE tools are more effectively used in software development. These includes structure charts, data flow diagrams.

→ Second generation CASE tools covers the life cycle automation and application generation.

→ Seconde generation CASE tools repositories are supported by modern databases for efficient retrieval of information.

• ARCHITECTURE OF A CASE ENVIRONMENT:

→ An architecture of CASE environment is a collection of CASE tools and management to support software development process.

→ The CASE environment provides tool support for project management and planning, analysis, design, testing, maintenance.

→ CASE environment has four main components.
i.e. ① User interface – GUI.
② Tool management – toolkits are limited to programming configuration mangt
③ Object management – EX: text, design, project plan
④ Shared repository. – It is a centralized database

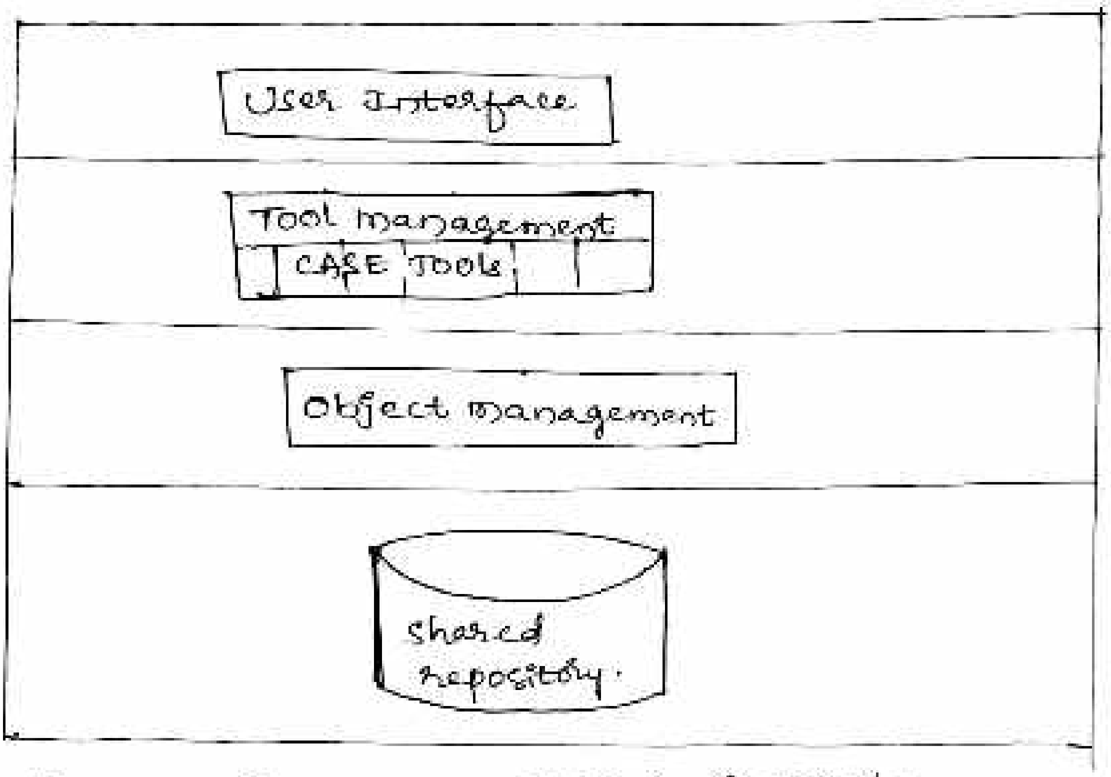

fig: Architecture of CASE environment.

* Important Questions:

1. Define Reliability? what are the reliability metrices?
2. Describe software quality management system (SQMS)?
3. Explain breefly SEI-CMM?
4. What are the principles of ISO 9000:2008.
5. Explain briefly ISO 9000 standard.
6. What are the advantages and disadvantages of CASE tools?
7. Discuss various tools of CASE environment?
8. Write a shortnotes on ⓐ Six Sigma
   ⓑ SQA plan

* ISO 9000 : 2008 standard prenciples are

1. customer focus
2. Leadership
3. Involvement of people
4. process approach
5. System approach to management
6. continuel improvement
7. Factual approach to decision making
8. Mutually beneficial supplier relationship.

* SQA (Software Quality Assurance) Activities:

The term 'assurance' means 'guarantee'.

1. Applying technical methods and tools
2. Conducting verification and validation
3. performing measurements
4. performing testing

* SQA plan:

1. purpose, scope, objectives of the SQA plan
2. Documents referenced in the plan
3. Tools and techniques and methodologies
4. Reviews and audits to be conducted.
5. code control
6. supplier controls.

* Six Sigma:

Key concepts of six sigma are

① critical to quality
② Defect
③ process capability
④ variation
⑤ stable operations
⑥ Design for six sigma

The six sigma "DMAIC" proces

D — Define
M — Measure
A — Analyse
I — Improve
C — Control.