Unit-6

Software Quality Management

TQM (Total Quality Management)

TQM is the integration of all functions and processes within an organization in order to achieve continuous improvement of the quality of goods and services.

The goal is customer satisfaction.

TQM is the enhancement to the traditional way of doing business.

It is a proven technique to guarantee survival in the world-class competition

Analyzing three words (TQM), we have:

1. Total- Made up of the whole
2. Quality- Degree of excellence a product or service provides
3. Management- Act, art, or manner of handling, controlling, directing, etc.

Therefore, TQM is the art of managing the whole to achieve the excellence.

Benefit of TQM

1) Greater customer loyalty
2) Market share improvement
3) Higher stock prices
4) Reduced service calls
5) Higher prices
6) Greater productivity
7) Improve quality
8) Employment participation

Principle of TQM

- Produce quality work for the first time and every time.
- Focus on the customer
- Have a strategic approach to improvement
- Improve continuously
- Encourage mutual respect and teamwork

Six Sigma

It is a set of techniques and tools for process improvement.

It seeks to improve the quality of process outputs by identifying and removing the causes of defects.

A six-sigma process is one in which 99.9999966% of the products manufactured are statistically expected to be free of defects.3.4 defect per million opportunities.

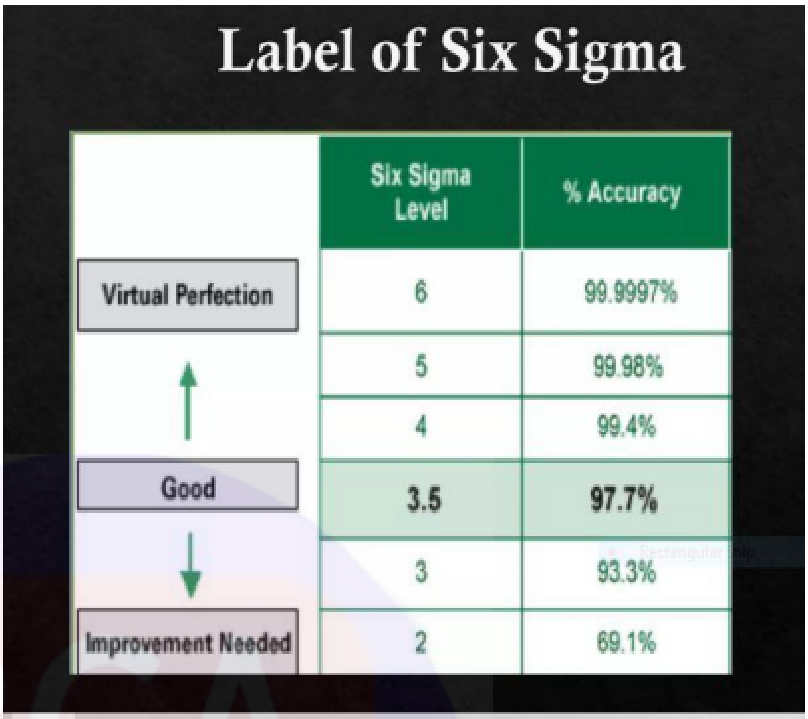
Sig sigma is a very clever way of branding and packaging many aspects of TQM (Total Quality Management).

1. **Define**: Clearly outline project goals and customer requirements.
2. **Measure**: Gather data to assess project performance and identify areas for improvement.
3. **Analyze**: Use statistical methods to understand root causes of issues.
4. **Improve**: Implement solutions to address identified problems.
5. **Control**: Establish processes to maintain improvements.
6. **Verify**: Validate the effectiveness of implemented changes and ensure they meet project objectives.

The **objectives of Six Sigma** in software project management are:

1. Improve quality and reduce defects.
2. Increase efficiency and productivity.

3. Enhance customer satisfaction.
4. Reduce variation and ensure consistency.
5. Optimize project performance.
6. Drive business growth and competitiveness.



Label of Six Sigma

	Six Sigma Level	% Accuracy
Virtual Perfection	6	99.9997%
	5	99.98%
	4	99.4%
Good	3.5	97.7%
	3	93.3%
Improvement Needed	2	69.1%

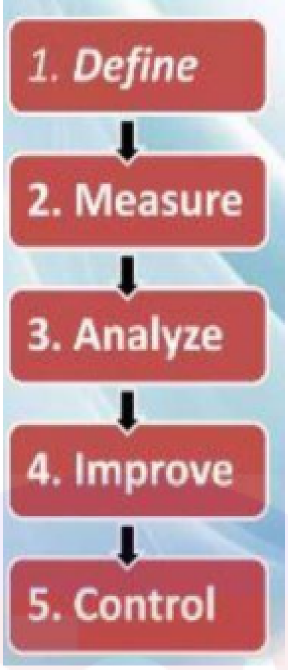
Six Sigma Methodology

1. DMAIC

Used for project aimed at improving an existing business process.

It has Five Phase

a) Define: define the system, the voice of customer, and their requirements, & the projects goal.
b) Measure: measure key aspects of the current process and collect relevant data.
c) Analyze: analyze the data to investigate and verify cause-effect relationships. Determine what the relationship are, and attempt to ensure that all factors have been considered.
d) Improve: improve the current process based upon data analysis using techniques such as design of experiments, to create a new, future state process.
e) Control: control the future state process to ensure that any deviations from target are corrected before they



2. DMADV

Used for projects aimed at creating new product or process design

It has Five Phase:

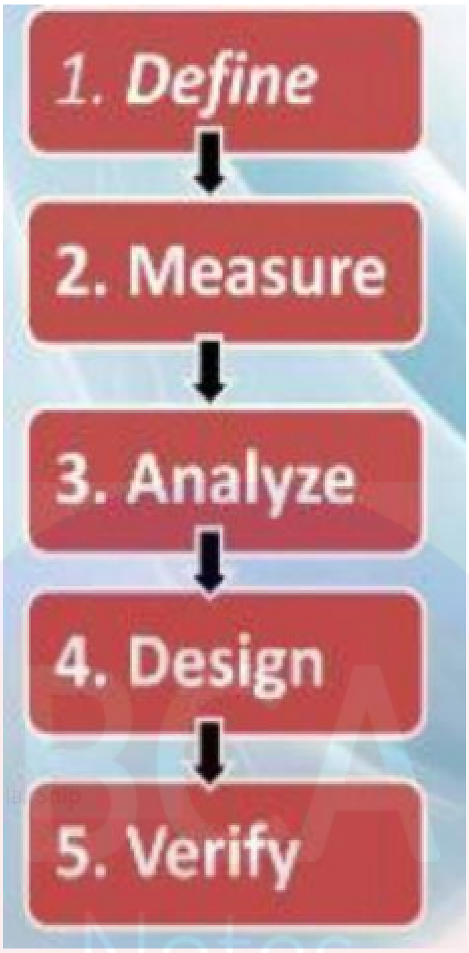
a. Define: define design goals that are consistent with customer demands and the enterprise strategy.

b. Measure: measure and identify characteristics that are critical to quality, product capabilities, production process capability, and risks.

c. Analyze: analyze to develop and design alternatives.

d. Design: design an improve alternatives

e. Verify: verify the design, set up pilot runs, implement the production process and hand it over to process owners.



Software Quality: Defining and importance of software Quality

Software quality management (SQM) is a management process that aims to develop and manage the quality of software in such a way so as to best ensure that the product meets the quality standards expected by the customer while also meeting any necessary regulatory and developer requirements. Software quality managers require software to be tested before it is released to the market, and they do this using a cyclical process-based quality assessment in order to reveal and fix bugs before release.

Software quality management activities are generally split up into three core components: quality assurance, quality planning, and quality control.

to help coordinate and align an organization's activities to uphold the customer and regulatory requirements while focusing on improving effectiveness and efficiency on a continuous basis.

How to measure software quality?

1. Correctness: Conformance to the specifications and user requirements.

2. Reliability: Performing expected function with required precision.

3. Efficiency: Number of resources required to execute required function.

4. Integrity: security features or security controls implemented in software.

5. Usability: Effort required to understand, learn and operate the software.

6. Maintainability: Effort required to maintain the software.

7. Testability: Effort required to test the software.

8. Flexibility: Effort required to make changes to the software.

9. Portability: Effort required to port software from one to another platform or to configuration.

10. Interoperability: Effort required to couple systems with one another.

Three Core Components

1. Quality Assurance

Established organizational quality standards

2. Quality Planning

Quality planning works at a more granular, project-based level, defining the quality attributes to be associated with the output of the project and how those attributes should be assessed

3. Quality Control

The quality control team tests and reviews software at its various stages to ensure quality assurance processes and standards at both the organizational and project level are being followed.

Importance of Software Quality

1. Saves time and money
2. Strengthen security
3. Brand reputation

4. User satisfaction

ISO9126

ISO 9126 standard was published in 1991 to tackle the question of the definition of software quality this 13 pages document was designed as foundation upon which further, more detailed standard could be built.

ISO 9126 identifies six software quality characteristics

1) Functionality: which covers the functions that a software product provides to satisfy user needs
2) Reliability: Which relates to the capability of the software to maintain its level of performance
3) Usability: Efforts need to use a software
4) Efficiency: physical resource used when a software is executed
5) Maintainability: Effort needed to the make changes to the software
6) Portability: Availability of the software to be transferred to a different environment.

ISO 9126 is an international standard that defines six main characteristics of software quality:

1. Functionality:

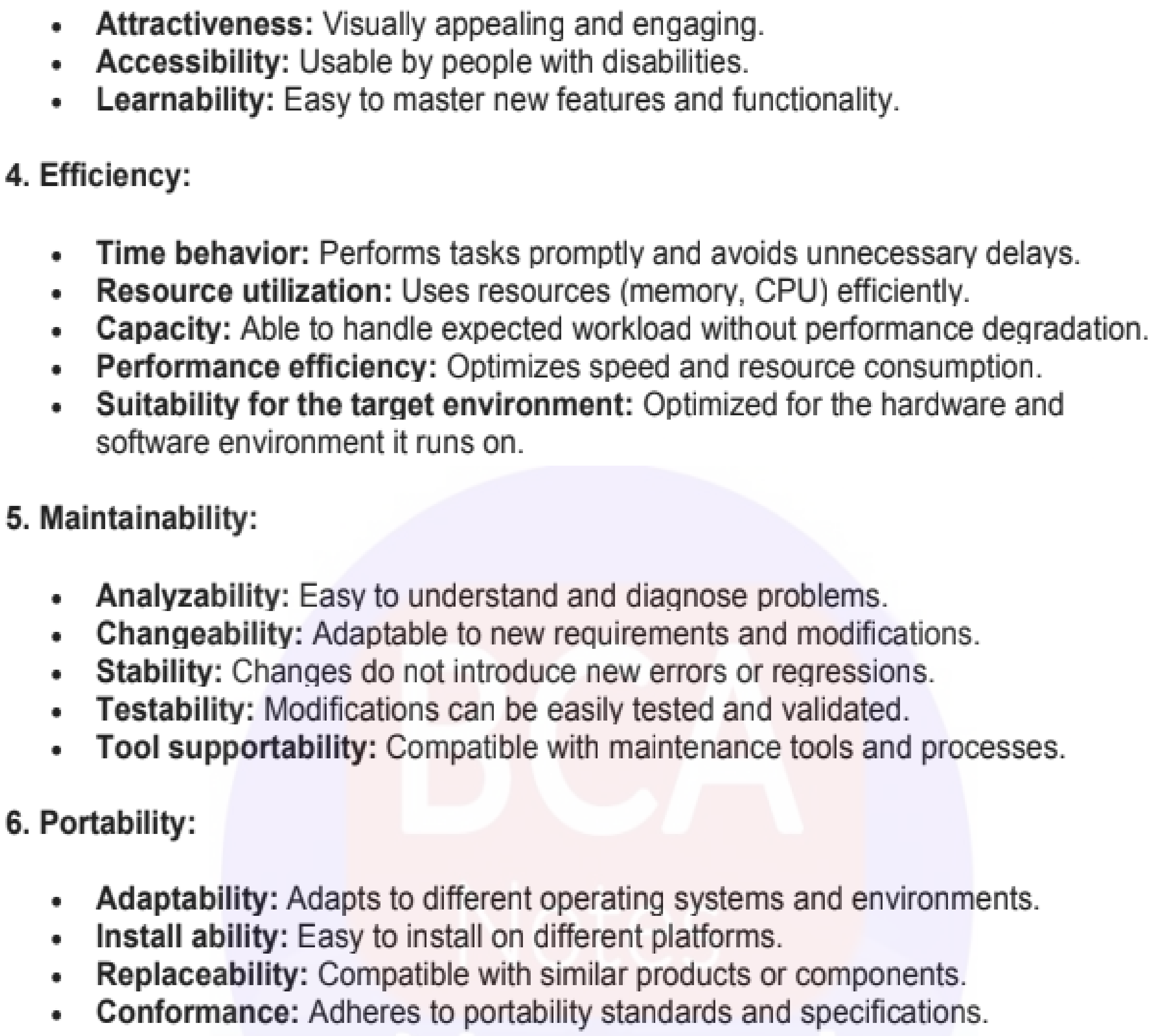
- **Suitability:** Provides adequate features and functions for intended use.
- **Accuracy:** Delivers correct and consistent results.
- **Interoperability:** Works effectively with other systems and components.
- **Compliance:** Adheres to relevant standards and regulations.
- **Security:** Protects data and assets from unauthorized access or harm.

2. Reliability:

- **Maturity:** Stable and operates as expected under normal conditions.
- **Availability:** Accessible and usable when needed.
- **Recoverability:** Can recover from failures quickly and gracefully.
- **Fault tolerance:** Continues to function despite minor errors.
- **Testability:** Can be easily and effectively tested.

3. Usability:

- **Understandability:** Clear and easy to learn and use.
- **Operability:** User interface is efficient and user-friendly.



- **Attractiveness:** Visually appealing and engaging.
- **Accessibility:** Usable by people with disabilities.
- **Learnability:** Easy to master new features and functionality.

4. Efficiency:

- **Time behavior:** Performs tasks promptly and avoids unnecessary delays.
- **Resource utilization:** Uses resources (memory, CPU) efficiently.
- **Capacity:** Able to handle expected workload without performance degradation.
- **Performance efficiency:** Optimizes speed and resource consumption.
- **Suitability for the target environment:** Optimized for the hardware and software environment it runs on.

5. Maintainability:

- **Analyzability:** Easy to understand and diagnose problems.
- **Changeability:** Adaptable to new requirements and modifications.
- **Stability:** Changes do not introduce new errors or regressions.
- **Testability:** Modifications can be easily tested and validated.
- **Tool supportability:** Compatible with maintenance tools and processes.

6. Portability:

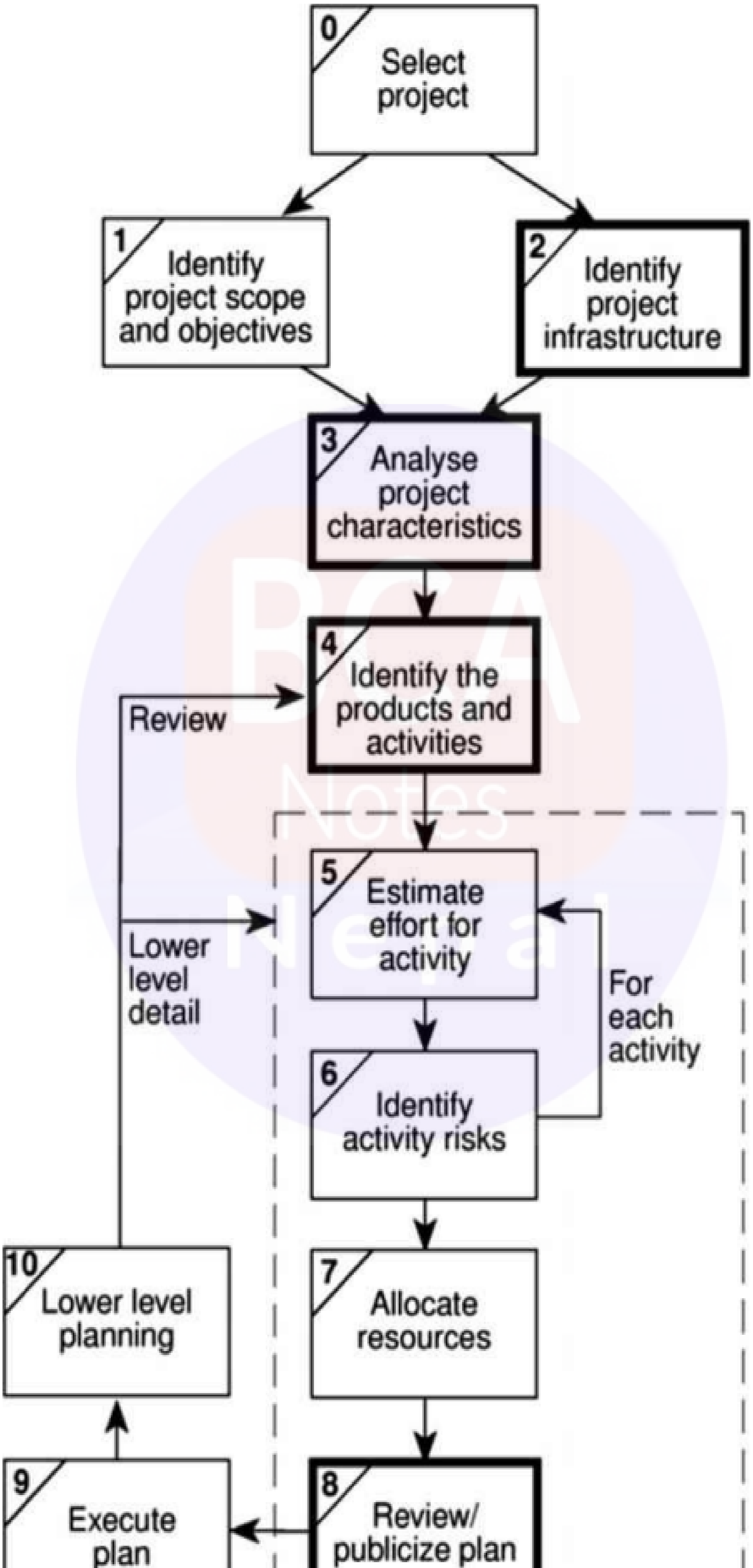
- **Adaptability:** Adapts to different operating systems and environments.
- **Install ability:** Easy to install on different platforms.
- **Replaceability:** Compatible with similar products or components.
- **Conformance:** Adheres to portability standards and specifications.

Place of Software Quality in software planning

In simple terms, the place of software quality in software planning means making sure the software turns out good. It's like setting a goal to make something really nice from the start and planning how to do it every step of the way. It's important because it helps avoid problems, keeps customers happy, and makes the project successful.

Advantages

1. **Saves Money:** Fewer delays, rework, and maintenance costs by catching issues early.
2. **Happy Customers:** Reliable, usable software leads to higher satisfaction and loyalty.
3. **Competitive Edge:** High-quality differentiates your product and builds trust.


0
Select project
1
Identify project scope and objectives
2
Identify project infrastructure
3
Analyse project characteristics
4
Identify the products and activities
Review
5
Estimate effort for activity
Lower level detail
For each activity
6
Identify activity risks
7
Allocate resources
10
Lower level planning
9
Execute plan
8
Review/ publicize plan