Software Project Management (SPM)

Course Code: CACS407
Year/ Semester: IV/VII

Compiled by Shishir Ghimire

Credit Hours: 3hrs

Unit - 06: Software Quality Management

Class Load : 5 Hrs

TABLE OF CONTENTS

Unit -6

Software Quality Management **5 Hrs**

TQM, Six Sigma, Software Quality: defining and importance software quality, ISO9126, Place of software quality in software planning.

Software Quality Management:

- Software Quality Management (SQM) is a comprehensive process that **ensures software products meet specified requirements**, and it involves planning, controlling, and improving software quality throughout the software development lifecycle.
- It includes :
  - Quality Planning
  - Quality Assurance (QA)
  - Quality Control (QC)
  - Continuous Improvement

TQM

6.1

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:
  - **Total** - Made up of the whole
  - **Quality** - Degree of excellence a product or service provides
  - **Management** - Act, art, or manner of handling, controlling, directing, etc.
- Therefore, TQM is the art of managing the whole to achieve the excellence.

TQM > Benefit of TQM :

- Greater customer loyalty
- Market share improvement
- Higher stock prices
- Reduced service calls
- Higher prices
- Greater productivity
- Improve quality
- Employment participation

TQM > 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

6.2

Six Sigma:

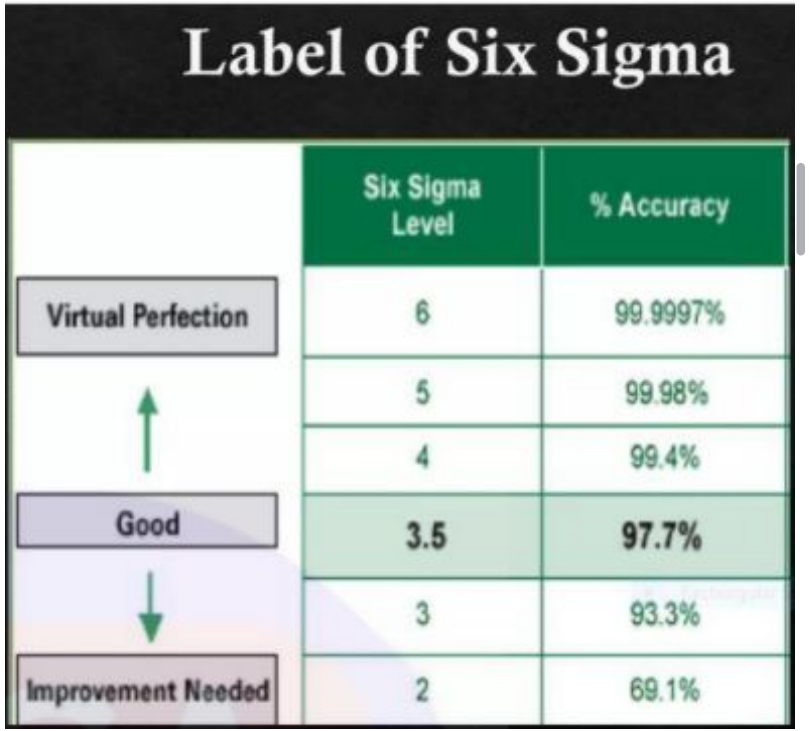
- It is a set of techniques and tools for **process improvement**.
- It seeks to improve the quality of process outputs by **identifying** and **removing** the causes of defects.
- A six-sigma process is one in which **99.9999966%** of the products manufactured are **statistically** expected to be **free of defects**. 3.4 defect per million opportunities.
- Six sigma is a very clever way of **branding** and **packaging** many aspects of TQM (Total Quality Management).

Six Sigma > Steps of Six Sigma :

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

Six Sigma > Objectives :

- Improve quality and reduce defects.
- Increase efficiency and productivity.
- Enhance customer satisfaction.
- Reduce variation and ensure consistency.
- Optimize project performance.
- Drive business growth and competitiveness.

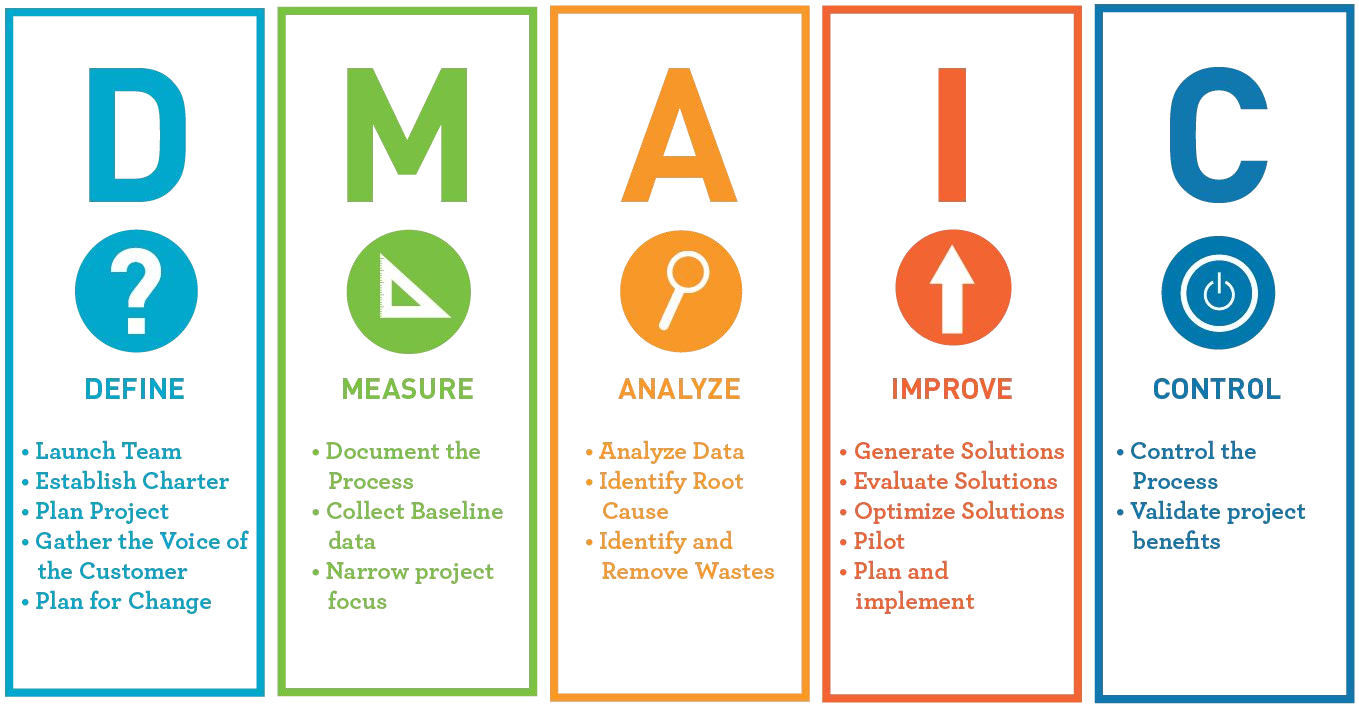


Six Sigma Methodologies > DMAIC :

- **Define:** Define the system, the voice of the customer, their requirements, and the project's goal.
- **Measure:** Measure key aspects of the current process and collect relevant data.
- **Analyze:** Analyze the data to investigate and verify **cause-effect relationships**. Determine what the relationships are and attempt to ensure that all factors have been considered.
- **Improve:** Improve the current process based on data analysis using techniques such as design of experiments to create a new, **future state process**.
- **Control:** Control the future state process to ensure that any deviations from the target are corrected before they occur.

Six Sigma Methodologies > DMAIC :

Six Sigma Methodologies > DMAIC :



Six Sigma Methodologies > DMADV :

DMADV is used for projects aimed at creating new product or process designs.

- **Define:** Define design goals that are consistent with customer demands and the enterprise strategy.
- **Measure:** Measure and identify characteristics that are critical to quality, product capabilities, production process capability, and risks.
- **Analyze:** Analyze to develop and design alternatives.
- **Design:** Design and improve alternatives.
- **Verify:** Verify the design, set up pilot runs, implement the production process, and hand it over to process owners.

Six Sigma Methodologies > DMADV :

DMAIC vs DMADV :

DMAIC	DMADV
The emphasis of DMAIC is more on correcting an existing process and reducing existing variation	DMADV is more about creating a process (Standardize) with an optimized design or "Doing it Right first time"
DMAIC is "Correction"	DMADV is "Prevention"
DMAIC uses more of Statistical tool and numerical/quantitative analysis to arrive at the solution	DMADV uses Qualitative tools: QFD (Quality Function Deployment), KANO Model etc
Six Sigma focuses on one or two CTQ, looks at processes, and aims to improve the CTQ performance	DFSS focuses on every single CTQ that matters, looks at products and services as well as the processes by which they are delivered, and aims to bring a large scale improvement
DMAIC projects often take short duration to fix a customer problem & process improvements	DMADV projects are often much larger and take longer, and are often based on a long term business need for new products /service

Software Quality

6.3

Software Quality:

- Software quality is a critical aspect of software development, ensuring that the software meets or exceeds user expectations and requirements.
- It encompasses a range of attributes, including functionality, reliability, usability, efficiency, maintainability, and portability.
- High-quality software is essential for delivering a positive user experience and achieving business goals.

Defining and importance of Software Quality

6.3.1

Software Quality Management (SQM):

- SQM is a **management process** aimed at **developing and managing** the quality of software to ensure the product meets customer expectations, regulatory standards, and developer requirements.
- **Definition:** Software quality managers use a cyclical process-based quality assessment to test software before release and identify and fix bugs.
- **Components:**
  - **Quality Assurance:** Activities to ensure that the software development process and products meet specified requirements.
  - **Quality Planning:** Identifying which quality standards are relevant to the project and determining how to satisfy them.
  - **Quality Control:** Monitoring and recording results to ensure that performance meets the required standards.

How to Measure Software Quality?

- **Correctness:** Conformance to the specifications and user requirements.
- **Reliability:** Performing the expected function with required precision.
- **Efficiency:** Number of resources required to execute the required function.
- **Integrity:** Security features or security controls implemented in the software.
- **Usability:** Effort required to understand, learn, and operate the software.
- **Maintainability:** Effort required to maintain the software.
- **Testability:** Effort required to test the software.
- **Flexibility:** Effort required to make changes to the software.
- **Portability:** Effort required to port software from one platform or configuration to another.
- **Interoperability:** Effort required to couple systems with one another.

Software Quality

- **Three Core Components:**
  - **Quality Assurance:** Established organizational quality standards.
  - **Quality Planning:** Works at a granular, project-based level to define the quality attributes of the project's output and how these attributes should be assessed.
  - **Quality Control:** Tests and reviews software at various stages to ensure that quality assurance processes and standards are being followed at both the organizational and project levels.
- **Importance of Software Quality:**
  - Saves time and money.
  - Strengthens security.
  - Maintains Brand Reputation
  - User Satisfaction

ISO 9126

5.3.2

ISO 9126

- **ISO 9126** standard was published in **1991** to tackle the question of the definition of software quality. This **13-page document** was designed as a foundation upon which further, more detailed standards could be built.
- **On March 1, 2011**, ISO/IEC 9126 was replaced by **ISO/IEC 25010:2011 Systems and software engineering** - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. Compared to 9126, "security" and "compatibility" **were added** as main characteristics.

ISO 9126


ISO 9126
Maintainability
Analysability
Changeability
Testability
Stability
Efficiency
Ressource utilisation
Time behaviour
Portability
Adaptability
Replaceability
Installability
Co-existence
Reliability
Fault tolerance
Recoverability
Maturity
Functionality
Regularity
Accuracy
Security
Suitability
Interoperability
Usability
Learnability
Understandability
Attractiveness
Operability
+ Compliance

ISO 9126

- ISO 9126 identifies **six software quality** characteristics:
  - **Functionality:** Covers the functions that a software product provides to satisfy user needs.
  - **Reliability:** Relates to the capability of the software to maintain its level of performance.
  - **Usability:** Efforts needed to use a software product.
  - **Efficiency:** Physical resources used when a software is executed.
  - **Maintainability:** Effort needed to make changes to the software.
  - **Portability:** Availability of the software to be transferred to a different environment.

ISO 9126

- **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.

ISO 9126

- **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.

ISO 9126

- **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.

ISO 9126

- **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.

ISO 9126

- **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.

ISO 9126

- **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 Planning

5.3.3

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

Place of Software Quality in Software Planning:

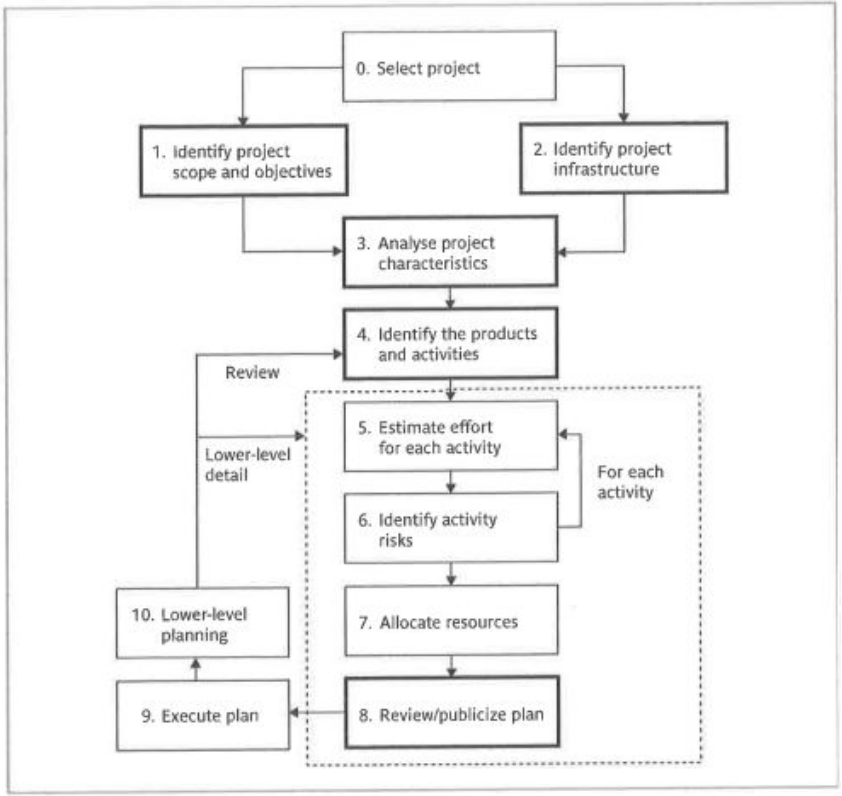


FIGURE 13.1 The place of software quality in Step Wise

Place of Software Quality in Software Planning:

- **Step 1: Identify project scope and objectives**: Define the project scope and objectives, ensuring they include the qualities of the application to be delivered.
- **Step 2: Identify project infrastructure:** Establish installation standards and procedures related to quality (activity 2.2).
- **Step 3: Analyse project characteristics:** Evaluate the project and its application for any special quality requirements (activity 3.2). For instance, safety-critical applications may require additional activities like n-version development to cross-check outputs for discrepancies.
- **Step 4: Identify the products and activities:** Define entry, exit, and process requirements for each activity to ensure they meet quality standards.
- **Step 8: Review and publicize plan:** Review the overall quality aspects of the project plan and communicate them to all stakeholders.

THANKS!

Do you have any questions?

theciceerguy@p_m.me

9841893035

ghimires.com.np