Software maintenance

Software maintenance is widely accepted part of SDLC now a days. It stands for all the modifications and updations done after the delivery of software product. There are number of reasons, why modifications are required, some of them are briefly mentioned below:

- **Market Conditions** - Policies, which changes over the time, such as taxation and newly introduced constraints like, how to maintain bookkeeping, may trigger need for modification.
- **Client Requirements** - Over the time, customer may ask for new features or functions in the software.
- **Host Modifications** - If any of the hardware and/or platform (such as operating system) of the target host changes, software changes are needed to keep adaptability.
- **Organization Changes** - If there is any business level change at client end, such as reduction of organization strength, acquiring another company, organization venturing into new business, need to modify in the original software may arise.

Types of maintenance

In a software lifetime, type of maintenance may vary based on its nature. It may be just a routine maintenance tasks as some bug discovered by some user or it may be a large event in itself based on maintenance size or nature. Following are some types of maintenance based on their characteristics:

- **Corrective Maintenance** - This includes modifications and updations done in order to correct or fix problems, which are either discovered by user or concluded by user error reports.
- **Adaptive Maintenance** - This includes modifications and updations applied to keep the software product up-to date and tuned to the ever changing world of technology and business environment.

- **Perfective Maintenance** - This includes modifications and updates done in order to keep the software usable over long period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance.
- **Preventive Maintenance** - This includes modifications and updations to prevent future problems of the software. It aims to attend problems, which are not significant at this moment but may cause serious issues in future.

Cost of Maintenance

Reports suggest that the cost of maintenance is high. A study on estimating software maintenance found that the cost of maintenance is as high as 67% of the cost of entire software process cycle.

On an average, the cost of software maintenance is more than 50% of all SDLC phases. There are various factors, which trigger maintenance cost go high, such as:

Real-world factors affecting Maintenance Cost

- The standard age of any software is considered up to 10 to 15 years.
- Older softwares, which were meant to work on slow machines with less memory and storage capacity cannot keep themselves challenging against newly coming enhanced softwares on modern hardware.
- As technology advances, it becomes costly to maintain old software.
- Most maintenance engineers are newbie and use trial and error method to rectify problem.
- Often, changes made can easily hurt the original structure of the software, making it hard for any subsequent changes.
- Changes are often left undocumented which may cause more conflicts in future.

Software-end factors affecting Maintenance Cost

- Structure of Software Program

- Programming Language
- Dependence on external environment
- Staff reliability and availability

Software quality assurance(SQA)

Software quality assurance(**SQA**) consists of a means of monitoring the software engineering processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards, such as ISO 9000 or a model such as CMMI.

SQA encompasses the entire software development process, which includes processes such as requirements definition, software design, coding, source code control, code reviews, software configuration management, testing, release management, and product integration. SQA is organized into goals, commitments, abilities, activities, measurements, and verifications.

Software Quality Factors

A software quality factor is a non-functional requirement for a software program which is not called up by the customer's contract, but nevertheless is a desirable requirement which enhances the quality of the software program.

Some software quality factors are listed here:

1. **Understandability** is possessed by a software product if the purpose of the product is clear. This goes further than just a statement of purpose - all of the design and user documentation must be clearly written so that it is easily understandable. This is obviously subjective in that the user context must be taken into account, i.e. if the software product is to be used by software engineers it is not required to be understandable to the layman.

2. A software product possesses the characteristic **completeness** to the extent that all of its parts are present and each of its parts is fully developed. This means that if the code calls a sub-routine from an external library, the software package must provide reference to that library and all required parameters must be passed. All required input data must be available.

3. A software product possesses the characteristic **conciseness** to the extent that no excessive information is present. This is important where memory capacity is limited, and it is important to reduce lines of code to a minimum. It can be improved by replacing repeated functionality by one sub-routine or function which achieves that functionality. It also applies to documents.

4. A software product possesses the characteristic **portability** to the extent that it can be operated easily and well on computer configurations other than its current one. This is particularly important with PC applications where, for example, a product is expected to work on all 80486 processors.

5. A software product possesses the characteristic maintainability to the extent that it facilitates updating to satisfy new requirements. Thus the software product which is maintainable should be **well-documented,** not complex, and should have spare capacity for memory usage and processor speed.

6. A software product possesses the characteristic **testability** to the extent that it facilitates the establishment of acceptance criteria and supports evaluation of its performance. Such a characteristic must be built-in during the design phase if the product is to be easily testable - a complex design leads to poor testability.

7. A software product possesses the characteristic usability to the extent that it is **convenient and practicable** to use. This is affected by such things as the human-computer interface. The component of the software which has most impact on this is the graphical user interface (GUI).

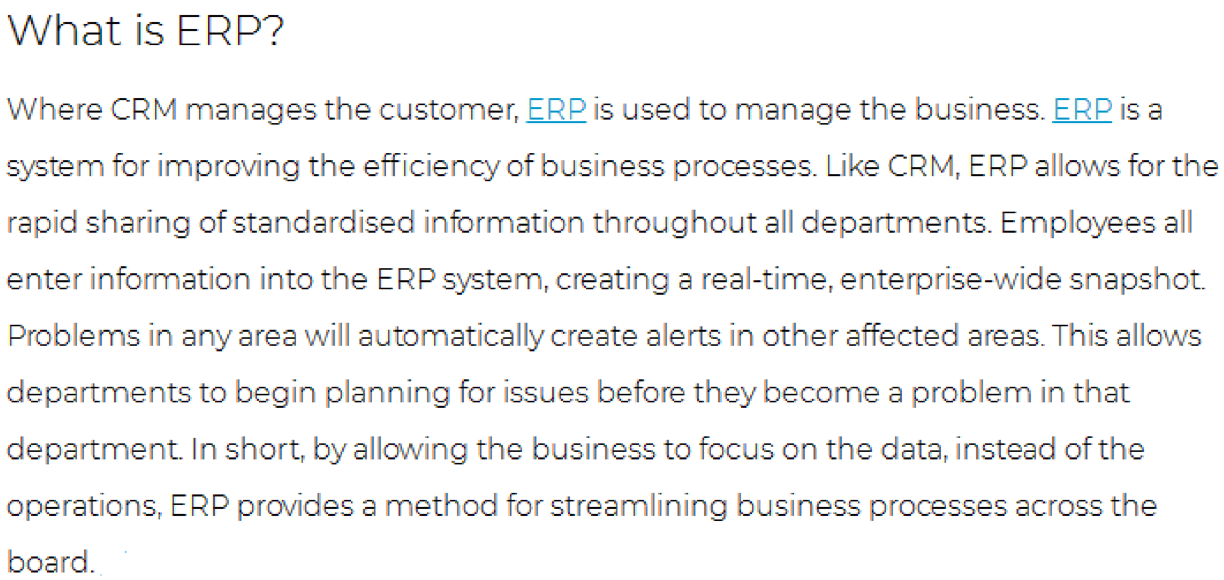
8. A software product possesses the characteristic **reliability** to the extent that it can be expected to perform its intended functions satisfactorily. This implies a time factor in that a reliable product is expected to perform correctly over a period of time. It also encompasses environmental considerations in that the product is required to perform correctly in whichever conditions it finds itself - this is sometimes termed robustness.
9. **Efficiency:** Fulfillment of purpose without waste of resources, such as memory, space and processor utilization, network bandwidth, time, etc.

10. **Security:** Ability to protect data against unauthorized access and to withstand malicious or inadvertent interference with its operations. Besides the presence of appropriate security mechanisms such as authentication, access control and encryption, security also implies resilience in the face of malicious, intelligent and adaptive attackers.

What is CRM?

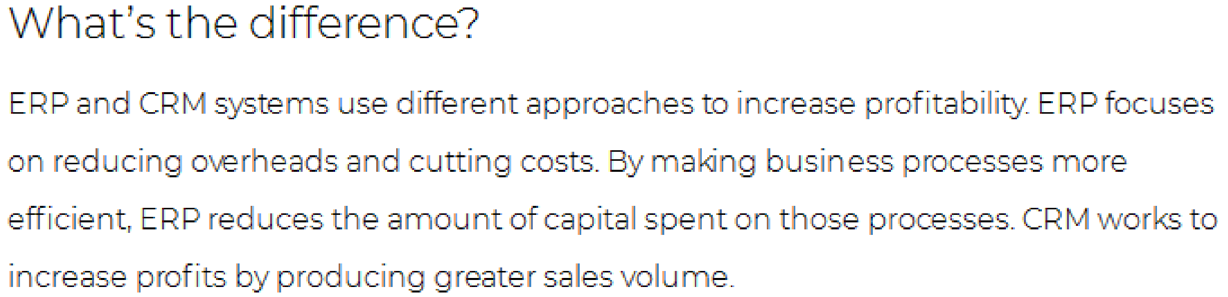
CRM at its simplest is systems and processes for managing a company's interactions with current and potential customers. When we talk about CRM we usually are talking about CRM Software. CRM software is used to organise, automate and synchronise sales, marketing and customer service.

CRM has developed to include all areas of the customer experience, keeping the customer happy and in turn keeping them loyal and more valuable to your business. It is the process of identifying potential leads/prospects, nurturing them and guiding them through the sales process to close the business. Once they are a customer it is ensuring that you maintain that relationship and encourage repeat business – either more frequent orders or higher value.



What is ERP?

Where CRM manages the customer, ERP is used to manage the business. ERP is a system for improving the efficiency of business processes. Like CRM, ERP allows for the rapid sharing of standardised information throughout all departments. Employees all enter information into the ERP system, creating a real-time, enterprise-wide snapshot. Problems in any area will automatically create alerts in other affected areas. This allows departments to begin planning for issues before they become a problem in that department. In short, by allowing the business to focus on the data, instead of the operations, ERP provides a method for streamlining business processes across the board.



What's the difference?

ERP and CRM systems use different approaches to increase profitability. ERP focuses on reducing overheads and cutting costs. By making business processes more efficient, ERP reduces the amount of capital spent on those processes. CRM works to increase profits by producing greater sales volume.

MRP (Material Requirements Planning) and MRP II (Manufacturing Resource Planning) are systems that control production and inventory. This means they are usually only utilized by the purchasing, production, and delivery departments.
Because MRP systems are exclusive to manufacturing operations, people that use it are generally in that department of a business. This could include a warehouse manager checking on the inventory of necessary raw materials, a warehouse worker checking on lead times or a production planning specialist overseeing the entire operation.

The point is that the types of people who use MRP are much more limited because the included services are only related to manufacturing. A more diverse group of users might benefit from ERP because of its range of provided workflows. ERP is a more expensive option compared to MRP.

ERP MRP CRM

ERP is the acronym for **Enterprise Resource Planning**, **MRP** stands for **Materials Requirement Planning** and **CRM** stands for Customer Relationship Management. ERP is used to plan the different processes in the organisation. MRP is specially designed for the manufacturing industry. CRM is acronym for **customer relationship management** and is used for maintaining healthy relationship with the customer. CRM software is used for managing and analyzing the customers interaction.

An **ERP (Enterprise Resource Planning)** has the potential to improve all your business operations by

1) facilitating useful communication between departments.

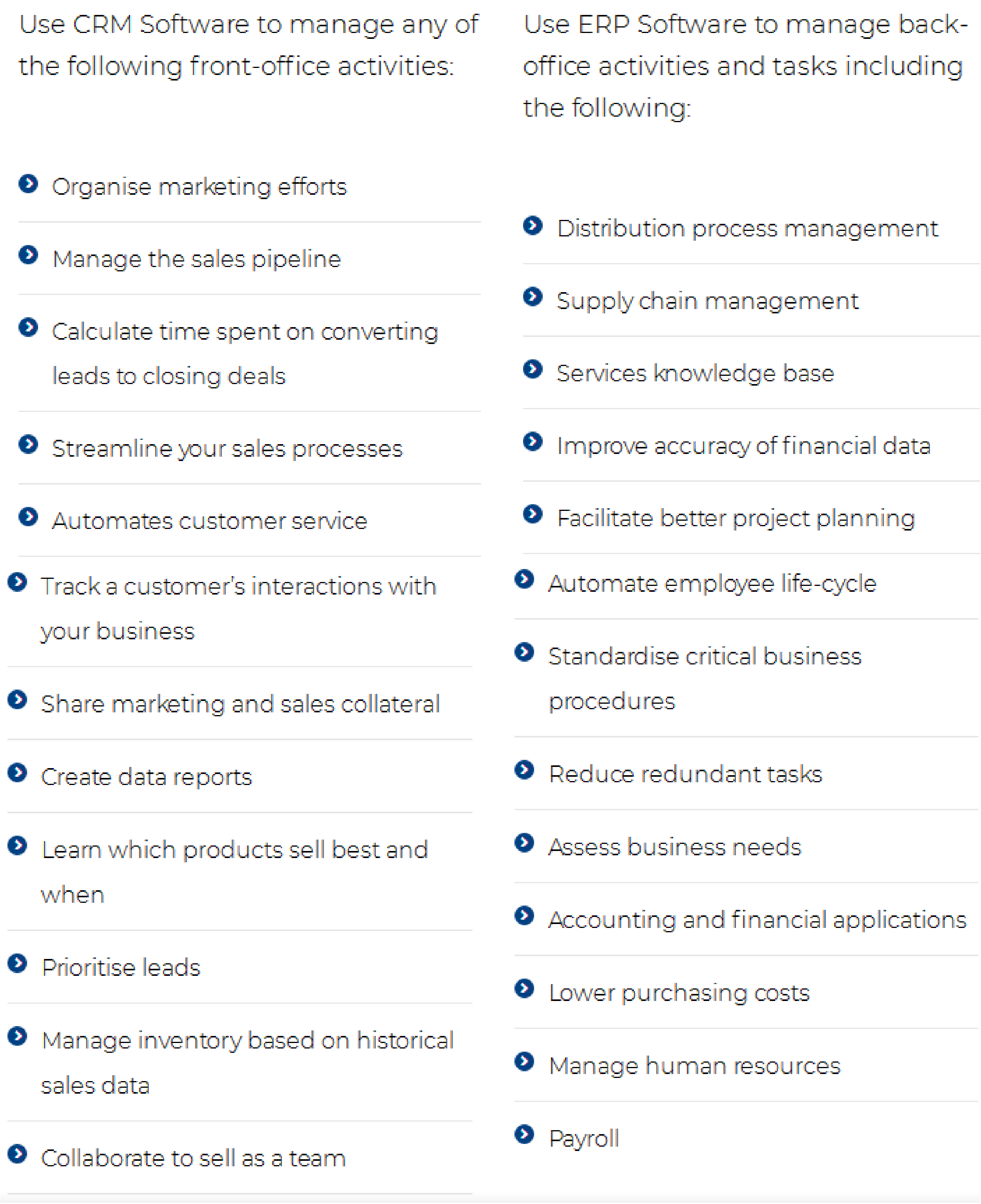
2) Making your entire system (organization) more responsive to the needs of all your stake holders.

An **ERP system** integrates all functions across a company to a single computer system, having a Centralized Data base, that can serve all those functions' specific needs.

Enterprise Resource Planning is used to organize and schedule the programs that are to be done in the industry. **MRP system** ensures that material is available for production, to ensure that there is always material available in stock, you can also plan manufacturing activities, delivery schedules and purchasing activities. From these definitions it is clear that MRP is the shorter version of ERP and the functions performed by CRM are not performed by MRP, hence ERP performs all the functions of MRP and CRM as well.

ERP has been extended to form OSM (Operating system for Manufacturing) as there are all the modules included in OSM so there is no need of ERP, MRP and CRM separately.

ERP is the system manager of the company, all the resources which are being used by the company should be properly utilized and there should be no wastage.



Use CRM Software to manage any of the following front-office activities:

- Organise marketing efforts
- Manage the sales pipeline
- Calculate time spent on converting leads to closing deals
- Streamline your sales processes
- Automates customer service
- Track a customer's interactions with your business
- Share marketing and sales collateral
- Create data reports
- Learn which products sell best and when
- Prioritise leads
- Manage inventory based on historical sales data
- Collaborate to sell as a team

Use ERP Software to manage back-office activities and tasks including the following:

- Distribution process management
- Supply chain management
- Services knowledge base
- Improve accuracy of financial data
- Facilitate better project planning
- Automate employee life-cycle
- Standardise critical business procedures
- Reduce redundant tasks
- Assess business needs
- Accounting and financial applications
- Lower purchasing costs
- Manage human resources
- Payroll

Supply chain management(SCM) software as a combination of warehouse management, transportation management, and the visibility, and optimization engines that address the search for the best source, delivery and inventory positions to keep the supply chain responsive and efficient. Whereas SCM systems are focused on the external movement and storage of goods, ERP systems are internally focused, with and execution modules for production, inventory, resources and financials, to name a few.

These two systems clearly have an intersection point when goods leave the plant on their way through the distribution chain and on to the customer, and perhaps when supplies, components and outsourced products leave the vendor or subcontractor on their way to the plant.

While SCM systems are unlikely to reach into the production space and attempt to control internal inventory and schedules, most ERP systems provide some level of visibility and control of warehouses and inbound and outbound goods. Entire functional areas -- customer order management, for example -- may be addressed in either or both ERP and SCM.

CMM

The Capability Maturity Model model is an organizational model that describes 5 evolutionary stages (levels) in which an organization manages its processes.

The thought behind the Capability Maturity Model, originally developed for software development, is that an organization should be able to absorb and carry its software applications. The model also provides specific steps and activities to get from one level to the next.

The 5 Stages of the Capability Maturity Model are:

- 1. Initial (processes are ad-hoc, chaotic, or actually few processes are defined)
- 2. Repeatable (basic processes are established and there is a level of discipline to stick to these processes) It is characteristic of processes at this level that some processes are repeatable, possibly with consistent results.
- 3. Defined (all processes are defined, documented, standardized and integrated into each other)

- 4. Managed (processes are measured by collecting detailed data on the processes and their quality)
- 5. Optimizing (continuous process improvement is adopted and in place by quantitative feedback and from piloting new ideas ands technologies)

What Is ISO 9001:2008 Certification?

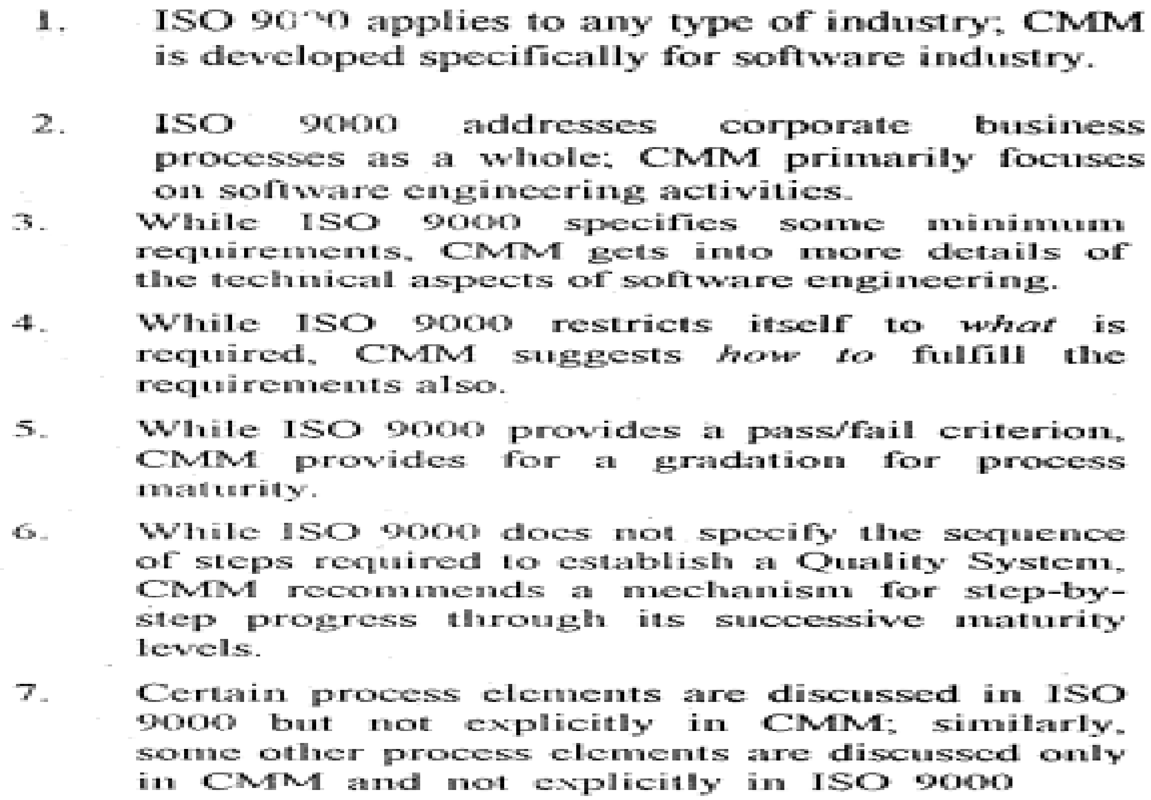
- While ISO9001 is not specific to the translation or localization industry, it does certify that there are well documented processes. This business certificate focuses on having a Quality Management System (QMS).
- ISO9001 certification demonstrates and monitors effective planning, operation and control of a company's processes and the implementation and continual improvement of its QMS. The ISO framework includes a stated quality policy, a quality manual and documented procedures, along with documented organization, implementation and record keeping.
- ISO9001 was set up for companies to focus on meeting customer needs, delivering what was promised to customers and improving overall quality of products and services. ISO 9001 looks for continuous improvement. When deficiencies or mistakes are identified, a process is set up to evaluate and address the shortcoming to continually improve the process.

Requirements of ISO 9001:2008

- ISO 9001: 2008 is an international standard related to quality management system, applicable to any organization from all types of business sectors and activities.
- ISO 9001:2008 is based on eight quality management principles (all fundamental to good business practice). When fully adopted, these principles can help improve your organizational performance:
- **Customer focus**: organizations depends on their customers, and therefore need to shape activities around the fulfillment of market need
- **Leadership**: is needed to provide unity of purpose and direction

- **Involvement of people**: creates an environment where people become fully involved in achieving the organization's objectives.
- **Process approach**: to achieve organizational objectives, resources and activities need to be managed as processes, with an understanding of how the outputs of one process affects the inputs of another
- **System approach to management**: the effectiveness and efficiency of the organization depends on a systemized approach to work activities
- **Continual improvement**: adopting this as a part of everyday culture is a key objective for an organization
- **Fact based decision-making**: effective decisions are based on the logical and intuitive analysis of data and factual information
- **Mutually beneficial supplier relationships**: such relationships will enhance theability to create value.

Major differences between ISO 9000 and CMM



1. ISO 9000 applies to any type of industry; CMM is developed specifically for software industry.
2. ISO 9000 addresses corporate business processes as a whole; CMM primarily focuses on software engineering activities.
3. While ISO 9000 specifies some minimum requirements, CMM gets into more details of the technical aspects of software engineering.
4. While ISO 9000 restricts itself to *what* is required, CMM suggests *how to* fulfill the requirements also.
5. While ISO 9000 provides a pass/fail criterion, CMM provides for a gradation for process maturity.
6. While ISO 9000 does not specify the sequence of steps required to establish a Quality System, CMM recommends a mechanism for step-by-step progress through its successive maturity levels.
7. Certain process elements are discussed in ISO 9000 but not explicitly in CMM; similarly, some other process elements are discussed only in CMM and not explicitly in ISO 9000