Risk management:

Risk management is concerned with identifying risks and drawing up plans to minimize their effect on a project.

Categories of risk:

- Project risks affect schedule or resources.
- Product risks affect the quality or performance of the software being developed.
- Business risks affect the organisation developing or procuring the software.

Risk may include the following:

- Experienced staff leaving the project and new staff coming in.

- Change in organizational management.

- Requirement change or misinterpreting requirement.

- Under-estimation of required time and resources.

- Technological changes, environmental changes, business competition.

The Risk Management Process:

The following activities involved in risk management process:

- **Risk Identification -** Identify project, product and business risks. Make note of all possible risks, which may occur in the project.
- **Risk Analysis -** Categorize known risks into high, medium and low risk intensity as per their possible impact on the project. Assess the likelihood and consequences of these **risks.**
- **Risk Planning -** Analyze the probability of occurrence of risks at various phases. Make plan to avoid or face risks. Attempt to minimize their side-effects of the risk.
- **Risk Monitoring -** Closely monitor the potential risks and their early symptoms. Also monitor the effects of steps taken to mitigate or avoid them.
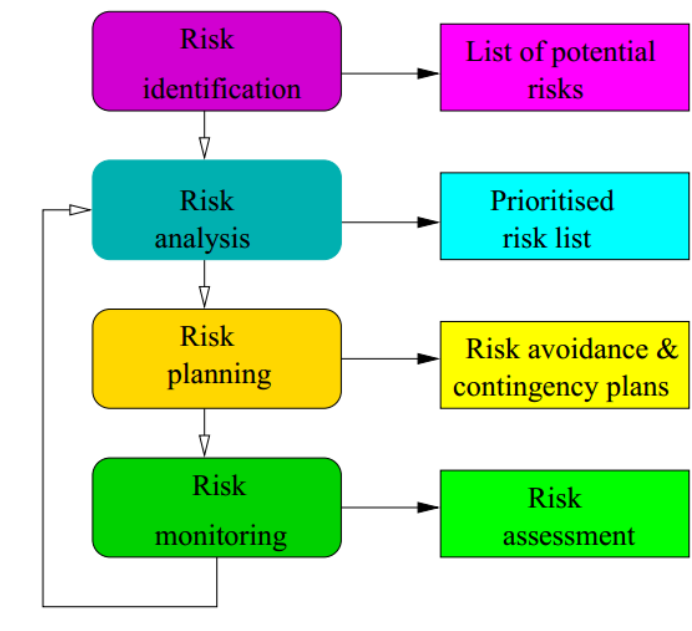
Manage Quality and Metrics:

- Quality is ultimately defined by the customer and represents how close the project and deliverables come to meeting the customer's requirements and expectations.

- The old adage about quality being in the eyes of the beholder is true - quality is ultimately measured by your customer. It is not up to the project team to determine the level of quality required for the project. The project team needs to understand the customer's requirements and expectations - and then meet those expectations.

- The purpose of quality management is to first understand the expectations of the customer in terms of quality and then put a proactive plan in place to meet those expectations.

Quality Standards for Software Products:

The most important quality features are:

- *Correctness*: The extent to which a program satisfies its specification and fulfills the customer's objectives.

- *Reliability*: The time dependent fulfillment of a given specification determines the reliability of a program.

- *User convenience:* is a collective term for the adequacy, Learnability, and robustness of a system.

- *Adequacy*: is concerned with the required input, the services program provides and the output of the program.

- *Learnability*: depends directly on the form of the user interfaces, and the clarity and simplicity of the user manual.

- *Robustness:* mean the property which weakness the effect of operating errors.

- *Maintainability:* means the ease with which errors can be localized and corrected and the ease with which the program functions can be altered or expanded.

- *Readability:* is depending on the programming style and its consistency, on the readability of the implementation language, on how structured the system is and most decisively on the quality of the documentation.

- *Expandability:* dependent on whether or not it's possible to insert the desired changes at the logically appropriate point in the program, with no undesired side effect.

- *Testability:* means how well the program facilitates testing of the program run and localization of program errors.

- *Efficiency:* the emphasis is on meeting people's requirements and use the hardware memory and CPU as carefully as possible.

- *Portability:* means the convenience with which the program can be implemented on various HW systems.