**Project Management** is the discipline of defining and achieving targets while optimizing the use of resources (time, money, people, materials, energy, space etc) over the course of a project. ( a set of activities of finite duration).

**Risk management** is the process of measuring or assessing risk and then developing strategies to manage the risk.

**Requirement management**  is the process of identifying, eliciting, documenting, analyzing, tracing, prioritizing and agreeing on requirements and then controlling change and communicating to relevant stakeholders.

**Change management** – how you made changes to the project.

Is the process of identifying, documenting, analyzing, prioritizing, and agreeing on changes to scope(project management) and then controlling changes and communicating to relevant stakeholders.

**Software Configuration management** – how you made changes to the software itself.

Is the process of identifying and documenting the scope itself, which is the software product underway including all sub-products and changes and enabling communication of these to relevant stakeholders.

**Project planning monitoring and control**

The purpose of [project planning](https://en.wikipedia.org/wiki/Project_planning) is to identify the scope of the project, [estimate](https://en.wikipedia.org/wiki/Estimation_in_software_engineering) the work involved, and create a [project schedule](https://en.wikipedia.org/wiki/Gantt_chart). Project planning begins with [requirements](https://en.wikipedia.org/wiki/Requirements_analysis) that define the software to be developed. The project plan is then developed to describe the tasks that will lead to completion.

The purpose of project monitoring and control is to keep the team and management up to date on the project's progress. If the project deviates from the plan, then the project manager can take action to correct the problem. Project monitoring and control involves status meetings to gather status from the team. When changes need to be made, [change control](https://en.wikipedia.org/wiki/Change_control) is used to keep the products up to date.

**Important issues of SW project management**

**Process improvement** – the SPI strategy transforms the existing approach to software development into something that is more focused, more repeatable, and more reliable ( in terms of the quality of the product produced and the timeliness of delivery)

**Requirements engineering –** refers to the process of defining, documenting and maintaining requirements and to the subfields of systems engineering and software engineering concerned with this process.

**Chapter 16 – Software Quality Assurance**

Overview

This chapter provides an introduction to software quality assurance. Software quality assurance (SQA) is the concern of every software engineer to reduce costs and improve product time-to-market. A Software Quality Assurance Plan is not merely another name for a test plan, though test plans are included in an SQA plan. SQA activities are performed on every software project. Use of metrics is an important part of developing a strategy to improve the quality of both software processes and work products.

Software Quality Assurance

* Umbrella activity applied throughout the software process
* Planned and systematic pattern of actions required to ensure high quality in software
* Responsibility of many stakeholders (software engineers, project managers, customers, salespeople, SQA group)

SQA Questions

* Does the software adequately meet its quality factors?
* Has software development been conducted according to pre-established standards?
* Have technical disciplines performed their SQA roles properly?

Quality Assurance Elements

* Standards – ensure that standards are adopted and follwed
* Reviews and audits – audits are reviews performed by SQA personnel to ensure hat quality guidelines are followed for all software engineering work
* Testing – ensure that testing id properly planned and conducted
* Error/defect collection and analysis – collects and analyses error and defect data to better understand how errors are introduced and can be eliminated
* Changes management – ensures that adequate change management practices have been instituted
* Education – takes lead in software process improvement and educational program
* Vendor management – suggests specific quality practices vendor should follow and incorporates quality mandates in vendor contracts
* Security management – ensures use of appropriate process and technology to achieve desired security level
* Safety – responsible for assessing impact of software failure and initiating steps to reduce risk
* Risk management – ensures risk management activities are properly conducted and that contingency plans have been established

SQA Tasks

* Prepare SQA plan for the project.
* Participate in the development of the project's software process description.
* Review software engineering activities to verify compliance with the defined software process.
* Audit designated software work products to verify compliance with those defined as part of the software process.
* Ensure that any deviations in software or work products are documented and handled according to a documented procedure.
* Record any evidence of noncompliance and reports them to management.

SQA Goals

* Requirements quality
* Ambiguity
* Completeness
* Volatility
* Traceability
* Model clarity
* Design quality
* Architectural integrity
* Component completeness
* Interface complexity
* Patterns
* Code quality
* Complexity
* Maintainability
* Understandability
* Reusability
* Documentation
* Quality control effectiveness
* Resource allocation
* Completion rate
* Review effectiveness
* Testing effectiveness

Formal SQA

* Assumes that a rigorous syntax and semantics can be defined for every programming language
* Allows the use of a rigorous approach to the specification of software requirements
* Applies mathematical proof of correctness techniques to demonstrate that a program conforms to its specification

Statistical Quality Assurance

1. Information about software defects is collected and categorized
2. Each defect is traced back to its cause
3. Using the Pareto principle (80% of the defects can be traced to 20% of the causes) isolate the "vital few" defect causes
4. Move to correct the problems that caused the defects in the “vital few”

Six Sigma Software Engineering

* *Define* customer requirements, deliverables, and project goals via well-defined methods of customer communication.
* *Measure* each existing process and its output to determine current quality performance (e.g. compute defect metrics)
* *Analyze* defect metrics and determine viral few causes.
* For an existing process that needs improvement
  + I*mprove* process by eliminating the root causes for defects
  + *Control* future work to ensure that future work does not reintroduce causes of defects
* If new processes are being developed
  + *Design* each new process to avoid root causes of defects and to meet customer requirements
  + *Verify* that the process model will avoid defects and meet customer requirements

Software Reliability

* Defined as the probability of failure free operation of a computer program in a specified environment for a specified time period
* Can be measured directly and estimated using historical and developmental data (unlike many other software quality factors)
* Software reliability problems can usually be traced back to errors in design or implementation.
* Measures of Reliability
* Mean time between failure (MTBF) = MTTF + MTTR
* MTTF = mean time to failure
* MTTR = mean time to repair
* Availability = [MTTF / (MTTF + MTTR)] x 100%

Software Safety

* Defined as a software quality assurance activity that focuses on identifying potential hazards that may cause a software system to fail.
* Early identification of software hazards allows developers to specify design features to can eliminate or at least control the impact of potential hazards.
* Software reliability involves determining the likelihood that a failure will occur, while software safety examines the ways in which failures may result in conditions that can lead to a mishap.

ISO 9000 Quality Standards

* Quality assurance systems are defined as the organizational structure, responsibilities, procedures, processes, and resources for implementing quality management.
* ISO 9000 describes the quality elements that must be present for a quality assurance system to be compliant with the standard, but it does not describe how an organization should implement these elements.
* ISO 9001:2000 is the quality standard that contains 20 requirements that must be present in an effective software quality assurance system.

SQA Plan

* Management section - describes the place of SQA in the structure of the organization
* Documentation section - describes each work product produced as part of the software process
* Standards, practices, and conventions section - lists all applicable standards/practices applied during the software process and any metrics to be collected as part of the software engineering work
* Reviews and audits section - provides an overview of the approach used in the reviews and audits to be conducted during the project
* Test section - references the test plan and procedure document and defines test record keeping requirements
* Problem reporting and corrective action section - defines procedures for reporting, tracking, and resolving errors or defects, identifies organizational responsibilities for these activities
* Other - tools, SQA methods, change control, record keeping, training, and risk management

Difference of lagging and leading indicators

Lagging indicators are typically “output” oriented, easy to measure but hard to improve or influence while leading indicators are typically input oriented, hard to measure and easy to influence.  
Let me illustrate this with a simple example: For many of us a personal goal is weight loss. A clear lagging indicator that is easy to measure. You step on a scale and you have your answer. But how do you actually reach your goal? For weight loss there are 2 “leading” indicators: 1. Calories taken in and 2. Calories burned. These 2 indicators are easy to influence but very hard to measure. When you order lunch in a restaurant the amount of calories is not listed on the menu. And if you are me, you have no clue how many calories you burn on a given day.