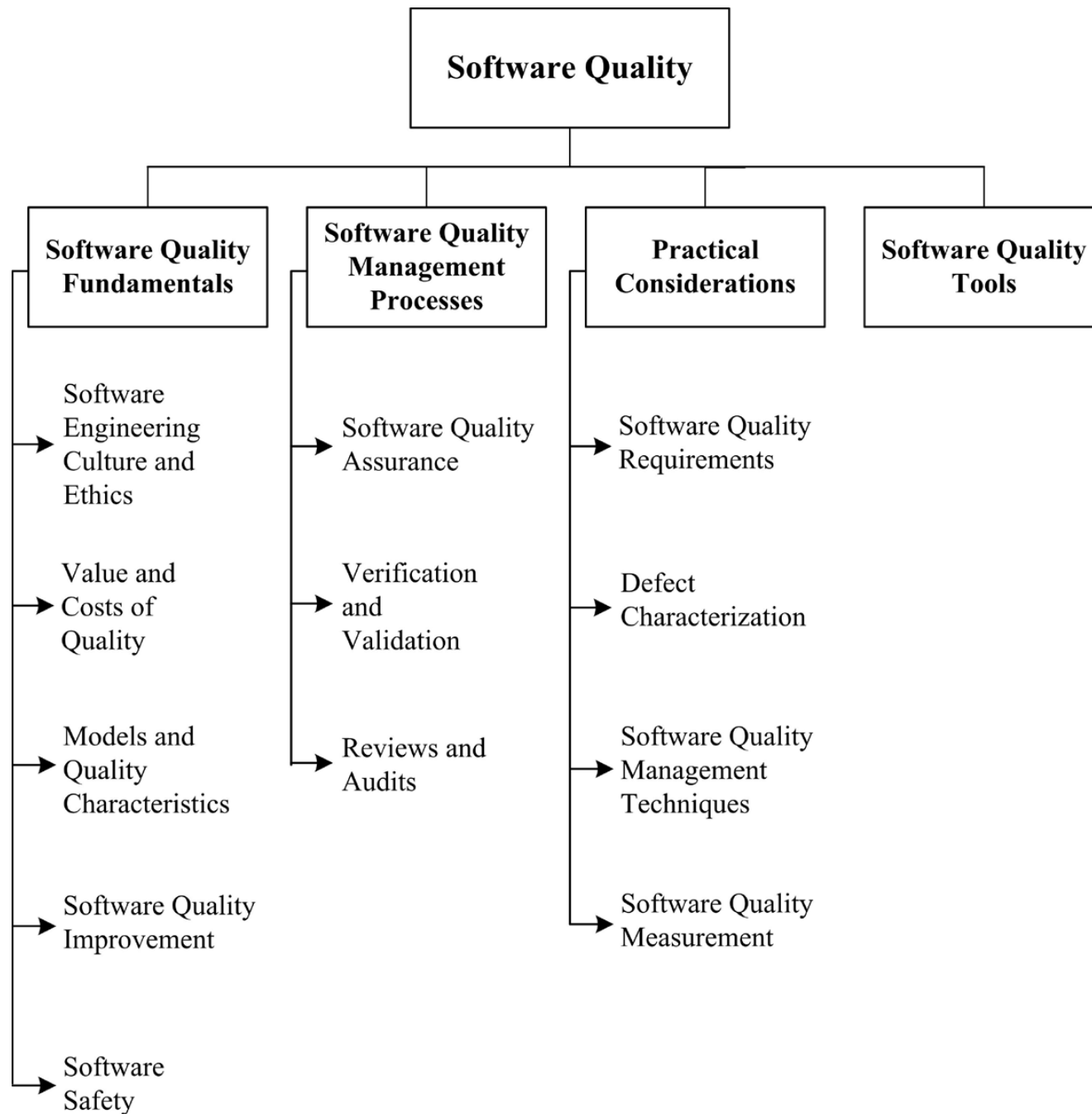
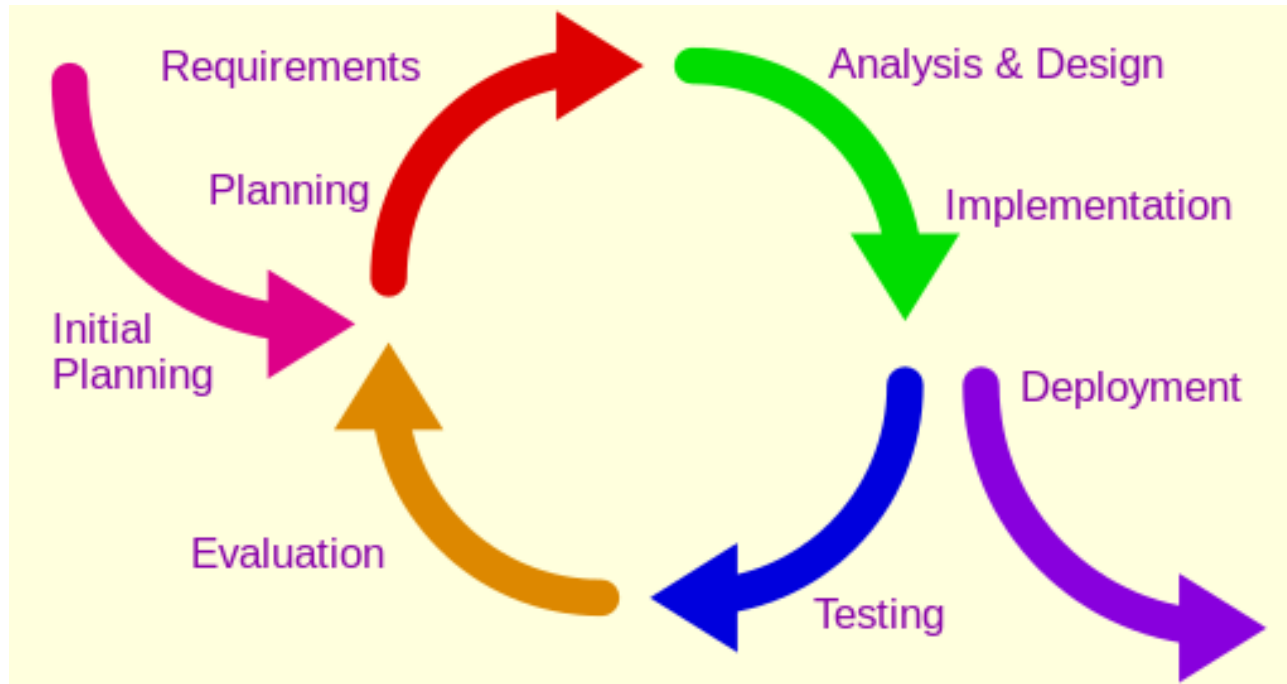


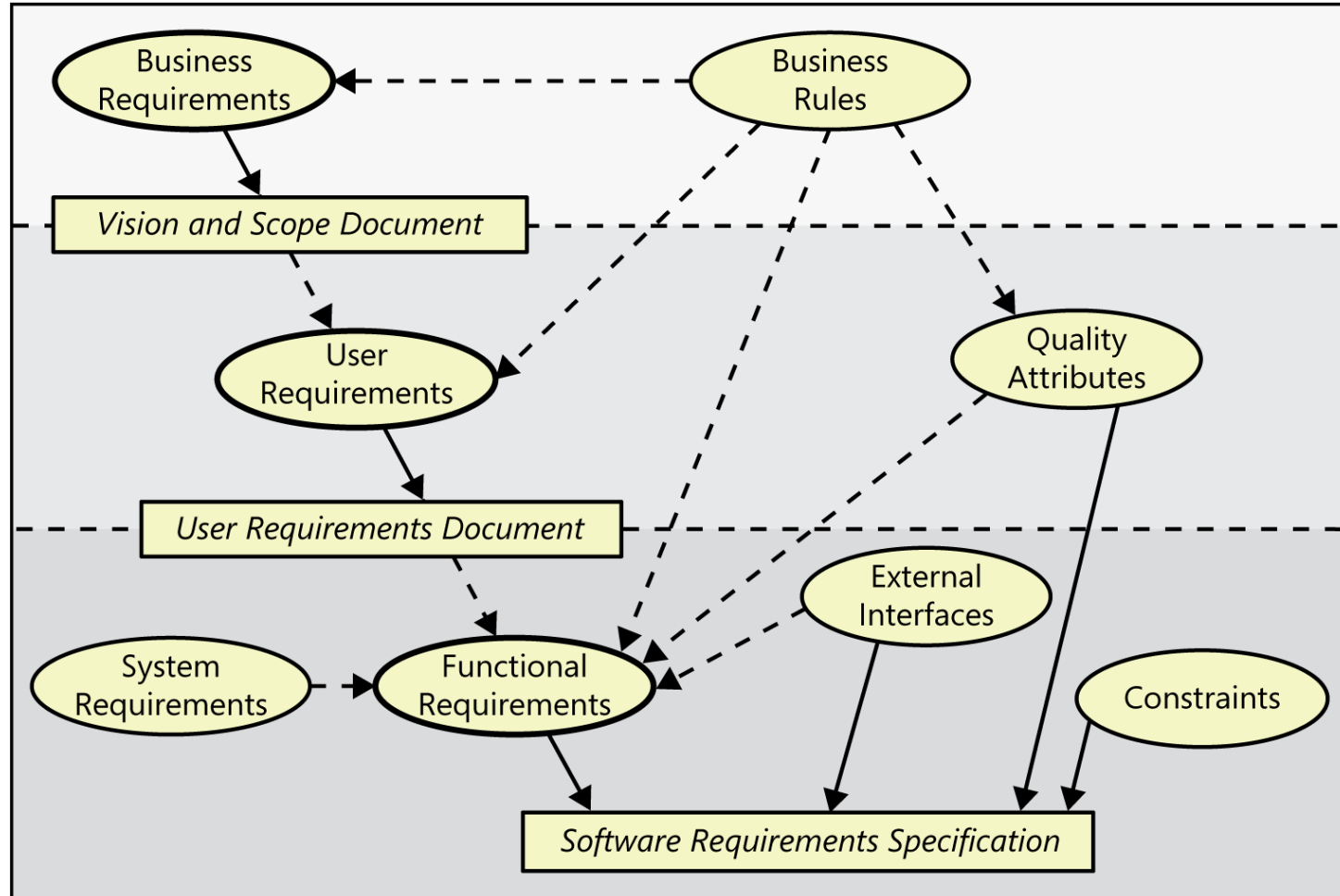
# **Software Quality**



# Iterative development model



# Relationships among several types of requirements



Solid arrows mean “are stored in”

Dotted arrows mean “are the origin of” or “influence.”

# Software quality

## ❑ Software quality :

- the capability of software product to satisfy **stated** and **implied needs** under specified conditions
- the degree to which a software product meets established requirements, how well the product performs its intended function
- a written or unwritten commitment to a known or unknown consumer in the market
- what a customer wants and is willing to pay for -- it determines quality

## ❑ **Quality assurance – QA** – process of application of quality concepts, characteristics and values to the software under development or maintenance

# Main quality concepts

- ❑ Software requirements define the required **quality attributes** of the software
- ❑ Software requirements influence the measurement methods and acceptance criteria for assessing the degree to which the software and related documentation achieve the desired **quality levels**
- ❑ Software quality is part of **software engineering culture**
  - the understanding tradeoffs among cost, schedule and quality
  - engineers accurately report information, conditions and outcomes related to quality
- ❑ Significant role of ethics and attitudes of software engineers

# Quality management

- ❑ **Quality management** ensures that an organization, product or service is **consistent**, is focused not only on product and service quality, but also on the means to achieve it
- ❑ Main **components**:
  - quality planning
  - quality assurance
  - quality control
  - quality improvement
- ❑ **ISO** – International Organization for Standardization (1987), ISO 9004:2009 document : guidelines for performance improvement
- ❑ **CMMI** – Capability Maturity Model Integration – (**SEI** - Software Engineering Institute) : methods of process assessment and improvement
  - product and service development (CMMI for Development)
  - service establishment, management, and delivery (CMMI for Services)
  - product and service acquisition (CMMI for Acquisition).

# Software quality attributes

- ❑ Software quality is achieved by conformance to **all** requirements regardless of what characteristic is specified or how requirements are grouped or named
- ❑ Quality Attributes contribute to the quality of a software design, including
  - “-ilities” : maintainability, portability, testability, usability...
  - “-nesses”: correctness, robustness ...
- ❑ Attributes discernible at runtime:
  - usability, availability, functionality, performance, security
- ❑ Attributes not discernible at runtime:
  - modifiability, portability, reusability, testability
- ❑ Attributes related to the architecture’s intrinsic qualities:
  - conceptual integrity, correctness, completeness



# Quality analysis: Reviews

- ❑ Software **design reviews** – informal and formalized **techniques to determine** the quality of design artifacts :
  - architecture reviews
  - design reviews and inspections
  - scenario-based techniques
  - requirements tracing
- ❑ Special review of software design to **evaluate security**
- ❑ Reviews of **aids** for installation, operation and usage of software

# Quality analysis and evaluation techniques

- ❑ **Static analysis:** formal or semiformal static (non executable) analysis of documents and source code:
  - evaluating a design (fault tree analysis)
  - vulnerability analysis (static analysis for security weaknesses)
  - formal design analysis – using mathematical models that allow to predicate the behavior and validate the performance of the software :
    - can be used to detect residual specification and design errors, perhaps caused by imprecision, ambiguity, and sometimes other kinds of mistakes
    - can be used in verification of crucial parts of critical systems, such as specific security and safety requirements
- ❑ **Simulation and prototyping:** **dynamic** techniques to evaluate a design (for performance simulation or feasibility prototypes)
- ❑ **Testing**
  - evaluate the testing itself: adequacy of plans, processes, and procedures and adequacy and accuracy of results
  - automated tests
  - tasking third-party organizations to perform testing or to monitor the test process

# Measures

- ❑ Measures can be used to assess or to quantitatively estimate various aspects of a software design: size, structure or quality
- ❑ Most measures depend on the approach used for producing design
- ❑ Measures are classified in two broad categories:
  - **Function-based (structured) design** measures:
    - obtained by analyzing functional decomposition
    - generally represented using a structure chart (a hierarchical diagram) on which various measures can be computed
  - **Object-oriented design** measures:
    - using a class diagram for representation of the design structure, on which various measures can be computed
    - measures on the properties of the internal content of each class can also be computed

# Value and costs of quality

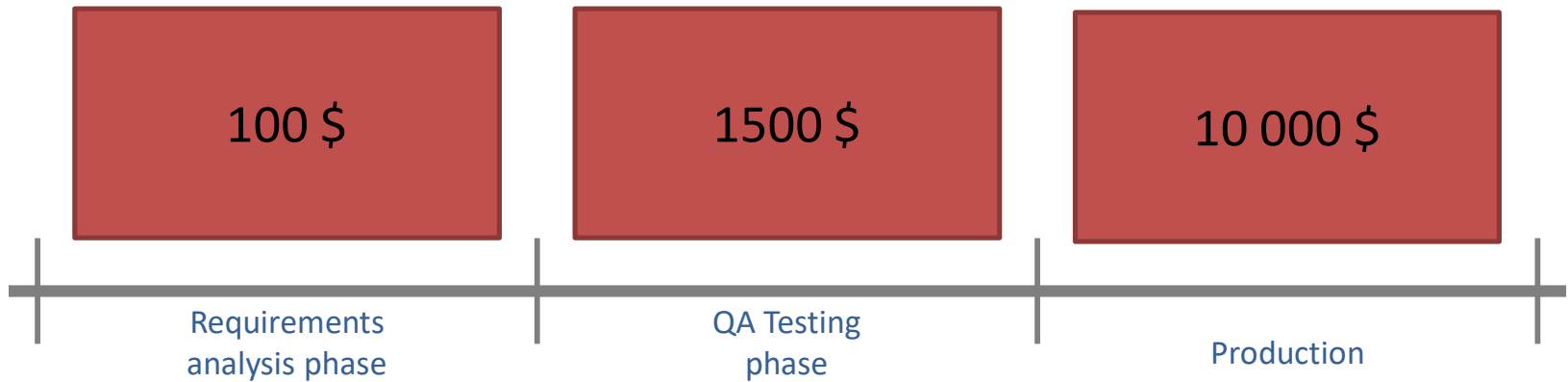
## ❑ **Costs of software quality** – a set of measurements derived from :

- economic assessment of software quality development
- maintenance processes

## ❑ **Categories** of software quality costs

- **Prevention costs** - are usually not specific to a project and are organization investments in :
    - quality infrastructure, quality tools
    - software process improvement efforts
    - training, audits and management reviews
  - **Appraisal costs** arise from project activities that find defects :
    - costs of reviews (design, peer)
    - costs of testing (software unit testing, software integration, system level testing, acceptance testing)
    - costs would be extended to subcontracted software suppliers
  - **Costs of internal failures** are those that are incurred to fix defects found during appraisal activities and discovered prior to delivery of the software product to the customer
  - **External failure costs** include activities to respond to software problems discovered after delivery to the customer
- ## ❑ **Measurements** of software quality costs are examples of **process measurements** that may be used **to infer characteristics** of a product

# Cost of fixing error on different phases



\* IAG Benchmark, 2008.

- ❑ **More than 80%** of errors originated on requirements analysis stage
- ❑ Fixing them eats up to **33% of project budget**

# Software process & product quality

- ❑ Direct bearing of SE **process quality** and software **quality management** on the quality of the software product
- ❑ SE process model influences the quality characteristics of software products and it isn't possible to completely distinguish process quality from product quality because **process outcomes include products**
- ❑ All software development processes are designed with elicited quality requirements in mind and may carry **additional development costs** if attributes such as **safety&security** are important
- ❑ **Work-products** (any artifact that is the outcome of a software process ):
  - a system/subsystem specification, a software requirements specification, a software design description, source code, software test documentation or reports
  - while treatments of quality are described in terms of **final** software and system performance, engineering practice requires that **intermediate** work-products relevant to quality be evaluated throughout the software engineering process.

# Software quality management processes

- ❑ **Software quality planning** – determining :
  - which quality standards are to be used
  - defining specific quality goals
  - estimating the effort and schedule of software quality activities
- ❑ **Software quality assurance** – activities, which define and assess the adequacy of software processes **to provide evidence that establishes confidence** that the software processes 1) are appropriate for and 2) produce software products of suitable quality for their **intended purposes**
- ❑ **Software quality control** – activities :
  - examine specific project artifacts (documents and executables) to determine whether they comply with standards established for the project
  - evaluate **intermediate** products as well as the **final** products

# Tasks & Techniques of software quality process

- ❑ Indicating
  - how software plans are being implemented
  - how well the intermediate and final products are meeting their specified requirements
- ❑ Assembling results of those tasks in reports for **management**, which may be used for **corrective** actions
- ❑ Improving product quality in the way of incorporating **risk analysis** and management techniques into the software life cycle processes



# Software quality assurance

- ❑ Objectivity of QA function with respect to the project – a key attribute of QA
- ❑ QA function may also be organizationally independent of the project: free from technical, managerial and financial pressures from the project
- ❑ QA has two aspects:
  - **product** assurance
  - **process** assurance
- ❑ QA activities and tasks are **specified** with their costs, resource requirements, objectives and schedule **in relation** to objectives in SE management, development and maintenance plans
- ❑ QA identifies :
  - measures, statistical techniques, procedures for problem reporting and corrective action
  - resources – tools, techniques, and methodologies
  - security for physical media
  - training and QA reporting and documentation
- ❑ QA identifies how its artefacts are checked and monitored to ensure adequacy and compliance

# QA practices: Verification & Validation

- ❑ **Verification** – attempt to ensure that the product is built correctly, in the sense that the output products of an activity **meet the specifications** imposed on them in previous activities
- ❑ **Validation** – attempt to ensure that **right product** is built—that is, the product fulfills its specific intended purpose
- ❑ V&V begin early in the development or maintenance phase
- ❑ V&V plan :
  - documents describe the various resources and their roles and activities as well as the techniques and tools to be used
  - addresses the management, communication, policies, and procedures of the V&V activities and their interaction as well as defect reporting and documentation requirements

# QA practices

## ❑ Reviews and audits

- examination of software engineering **artifacts** with respect to standards that have been established by the organization or project for those artifacts
- are broadly defined as **static** — no software programs or models are executed
- product assurance and process assurance **audits** are typically conducted by **QA personnel** who are independent of development teams

## ❑ Technical reviews — inspections, walkthrough and desk checking — examine engineering work-products; are conducted by engineering staff

- **Purpose** of a technical review is to evaluate a software product by a team to determine its suitability for its intended use and identify discrepancies from specifications and standards
- **Provides** management with evidence to confirm the technical status of the project

## ❑ Management reviews evaluate actual project results with respect to plans; are conducted by organizational or project management

- **Purpose** of a management review :
  - monitor progress
  - determine the status of plans and schedules
  - evaluate the effectiveness of management processes, tools and techniques.
- **Parameters** of management reviews are project cost, schedule, scope, and quality
- **Management** reviews evaluate decisions about corrective actions, changes in the allocation of resources, or changes to the scope of the project

# Defect characterization

Word **defect** is overloaded to refer to different types of anomalies at **all stages** of software development and maintenance

- ❑ **Error** — human action that produces an incorrect result; a mistake that person makes (also called **human error** )
- ❑ **Computational Error** — difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition
- ❑ **Defect** — imperfection or deficiency in a work product where that work product does not meet its requirements or specifications and needs to be either repaired or replaced
- ❑ **Fault** — defect in source code – an incorrect step, process, or data definition in computer program ( fault is the formal name of a **bug** )
- ❑ **Failure** — event in which a system or system component does not perform a required function within specified limits; a failure is produced when a fault is encountered by processor under specified conditions

# Software quality measurements

- ❑ Software quality measurement is used to support decision-making
- ❑ **Cost of quality measurements** processes is an issue frequently raised in deciding how a project development and maintenance group should be organized
- ❑ Generic models of cost are based on **when a defect is found** and how much effort it takes to fix the defect relative to finding the defect earlier in the development process.
- ❑ SQM data collected internally may give a better picture of cost within this project or organization

# Software quality measurements

- ❑ Mathematical and graphical techniques for interpretation of measures
  - descriptive statistics based (e.g., Pareto analysis, run charts, scatter plots, normal distribution)
  - statistical tests (e.g., binomial test – an exact test of the statistical significance of deviations from a theoretically expected distribution of observations into two categories)
  - trend analysis (e.g., control charts)
  - prediction (e.g., reliability models)
- ❑ QM includes measuring defect occurrences and applying statistical methods to understand the types of defects that occur most frequently:
  - may be used by software **process improvement** for determining methods to prevent, reduce, or eliminate their recurrence
  - **aid** in understanding trends, how well detection and containment techniques are working, and how well the development and maintenance processes are progressing

# Software quality management techniques

- ❑ **Static techniques** examine software documentation (including requirements, interface specifications, designs, and models) and software source code without executing the code
- ❑ **Dynamic techniques** involve executing the software code. Different kinds of dynamic techniques are performed throughout the development and maintenance of software. Generally, these are testing techniques as simulation and model analysis
- ❑ **Testing** – this party is not the developer, nor is it associated with the development of the product. Instead, it is an independent facility, usually accredited by some body of authority. Their purpose is to test a product for conformance to a specific set of requirements