



BITS Pilani presentation

BITS Pilani
Pilani Campus

Dr. N.Jayakanthan



BITS Pilani
Pilani Campus



SE ZG501

Software Quality Assurance and Testing

Lecture No. 1

Software Quality Assurance

SQA addresses the global challenge of the **improvement of software quality**.

It seeks to provide an overview of software quality assurance (SQA) practices for customers, managers, auditors, suppliers, and personnel responsible for software projects, development, maintenance, and software services.

In a globally competitive environment, **clients and competitors exert a great deal of pressure on organizations.**

Clients are increasingly demanding and require, among other things, software that is of **high quality, low cost, delivered quickly, and with impeccable after-sales support.**

To meet the **demand, quality, and deadlines**, the organization must use **efficient quality assurance practices** for their software activities.

Standards define ways to maximize performance but managers and employees are largely left to themselves to decide how to practically improve the situation.

They face several problems:

- increasing **pressure** to deliver quality products quickly
- increasing **size and complexity** of software and of systems
- increasing **requirements** to meet national, international, and professional standards
- **subcontracting and outsourcing**
- **distributed work teams**
- **ever changing platforms and technologies.**

INTRODUCTION



Software is developed, maintained, and used by people in a wide variety of situations.

Students ,enthusiasts , professionals address quality problems that arise in the software they are working with

The Guide to the **Software Engineering Body of Knowledge** (SWEBOK) [SWE 14] constitutes the first international consensus developed on **the fundamental knowledge required by all software engineers.**

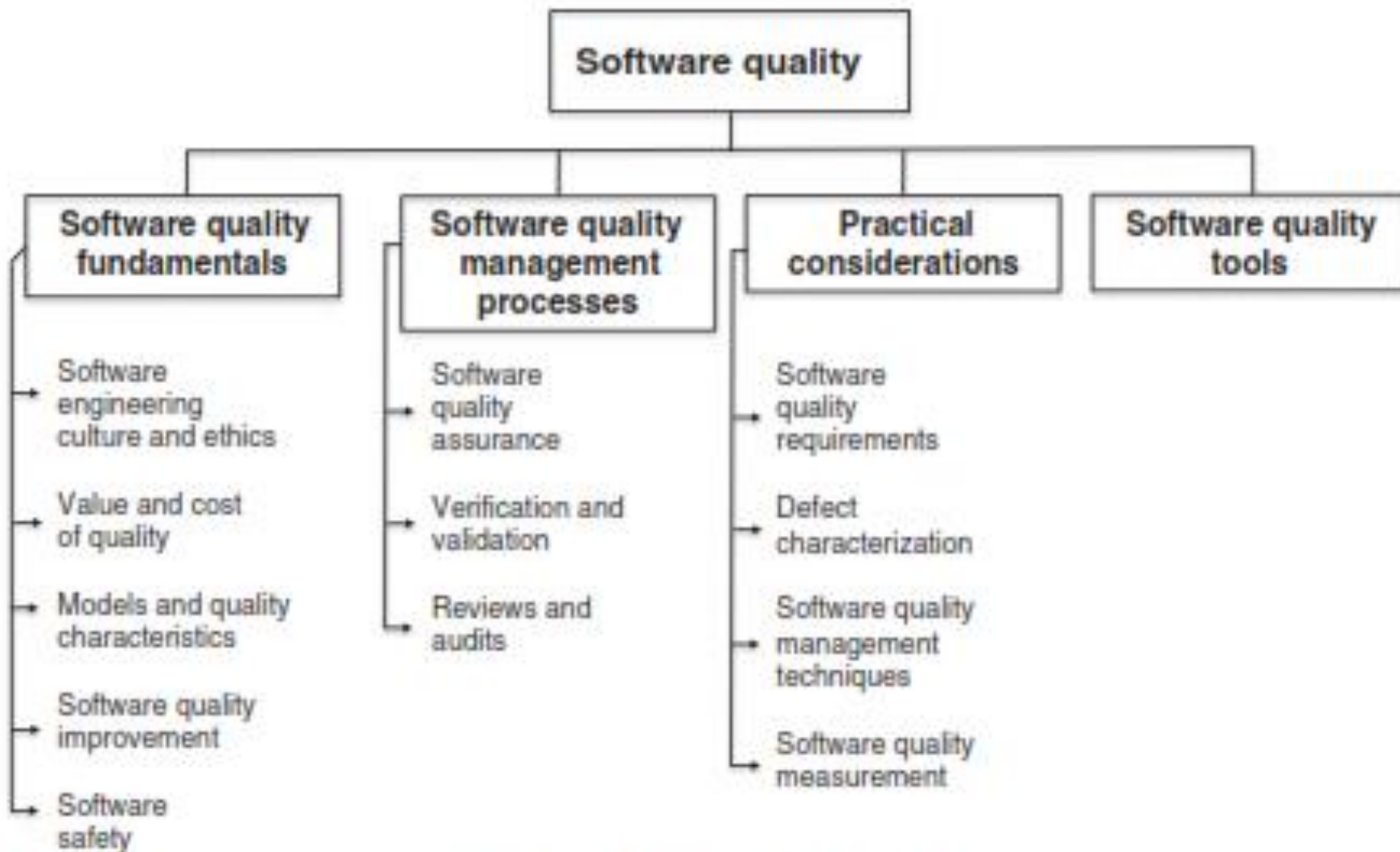


Figure 1.1 Software Quality in the SWEBOK® Guide [SWE 14].

DEFINING SOFTWARE QUALITY

“software” “software quality” and “software quality assurance”

Software [ISO 24765 [ISO 17a]]

- 1) All or part of the programs, procedures, rules, and associated documentation of an information processing system.**
- 2) Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.**

Software found in embedded systems is sometimes called **microcode or firmware**.

Firmware is present in **commercial mass-market products** and controls machines and devices **used in our daily lives**.

Firmware

Combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device.

SOFTWARE ERRORS, DEFECTS, AND FAILURES



- The system crashed during production.
- The designer made an error.
- After a review, we found a defect in the test plan.
- I found a bug in a program today.
- The system broke down.
- The client complained about a problem with a calculation in the payment report.
- A failure was reported in the monitoring subsystem.

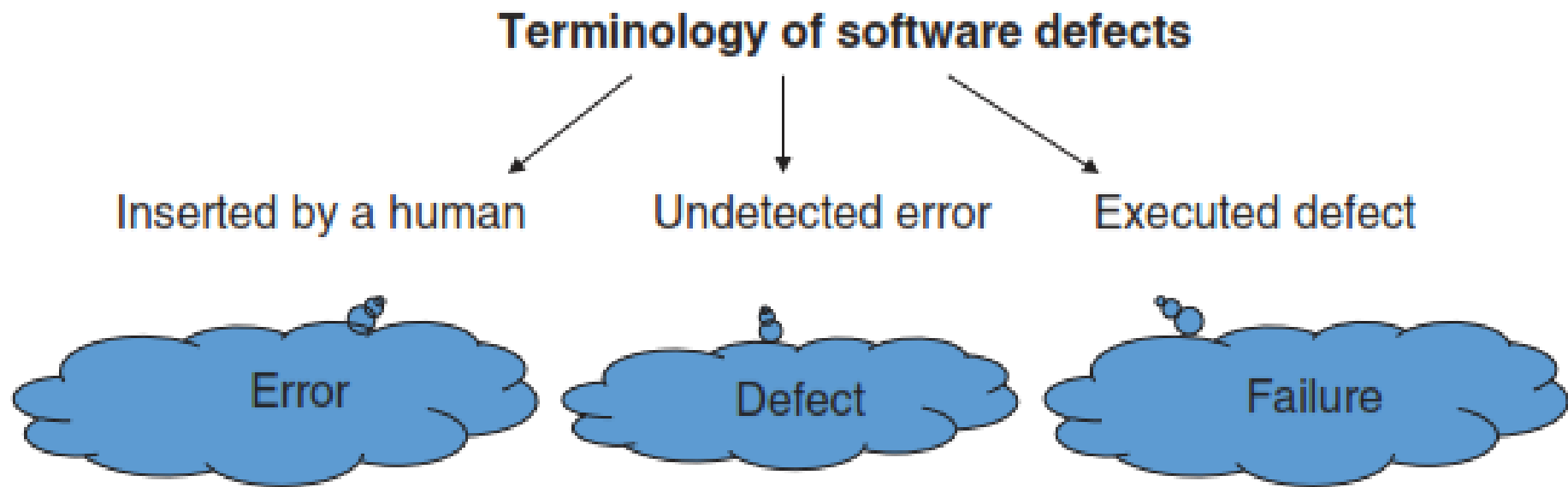


Figure 1.2 Terminology recommended for describing software problems.

9/9

0830

Antenna started

1000

stopped - antenna ✓

13.02.030 HP - MC

000 PRO 2

convd

Relays 6-2 in 030 failed special speed test
in relay

Relays changed

1100

Started Cosine Tape (Sine check)

1525

Started Multi-fader Test

1545

Relay to Panel F
(noth) in relay.

First actual case of bug being found.

Antenna started.

1700 closed down.

Relay
2142
030 030

A failure (synonymous with a crash or breakdown) is the execution (or manifestation) of a fault in the operating environment.

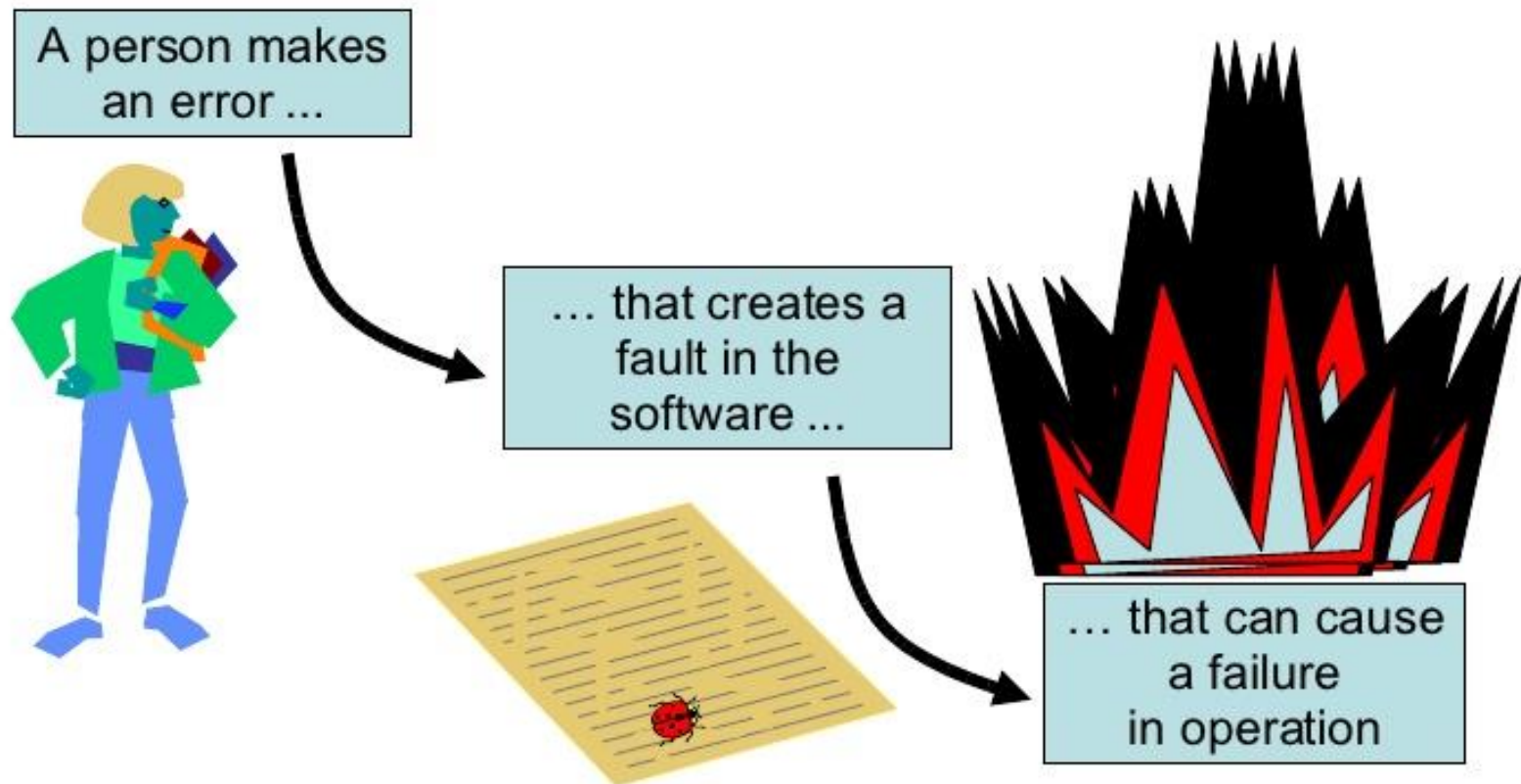
A failure is defined as the **termination of the ability of a component to fully or partially perform a function** that it was designed to carry out.

The origin of a failure lies with a **defect hidden**, that is, not detected by tests or reviews, in the system currently in operation.

Defects (synonym of faults) are human errors that were not detected during software development, quality assurance (QA), or testing.

An error can be found in the documentation, the software source code instructions, the logical execution of the code, or anywhere else in the life cycle of the system.

Error - Fault - Failure





Error, Defect, and Failure

Error

A human action that produces an incorrect result (ISO 24765) [ISO 17a].

Defect

- 1) A problem (synonym of fault) which, if not corrected, could cause an application to either fail or to produce incorrect results. (ISO 24765) [ISO 17a].
- 2) An imperfection or deficiency in a software or system component that can result in the component not performing its function, e.g. an incorrect data definition or source code instruction. A defect, if executed, can cause the failure of a software or system component (ISTQB 2011 [IST 11]).

Failure

The termination of the ability of a product to perform a required function or its inability to perform within previously specified limits (ISO 25010 [ISO 11i]).

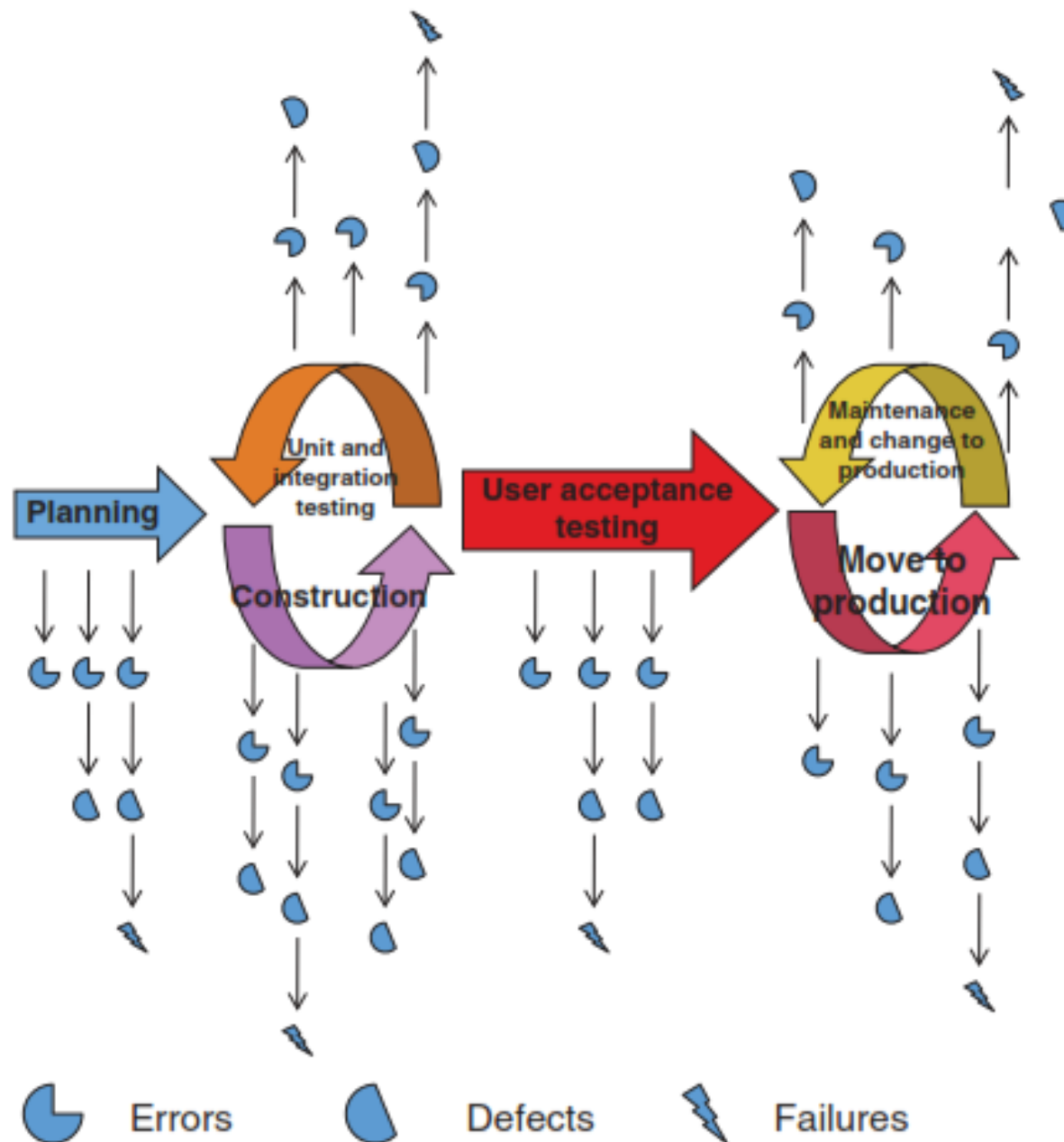


Figure 1.3 Errors, defects, and failures in the software life cycle.

Case 1:

A local pharmacy added a software requirement to its cash register to prevent sales of more than \$75 to customers owing more than \$200 on their pharmacy credit card.

The programmer did not fully understand the specification and created a sales limit of \$500 within the program. This **defect** never caused a failure since no client could purchase more than \$500 worth of items given that the pharmacy credit card had a limit of \$400.



Case 2: In 2009, a loyalty program was introduced to the clients of American Signature, a large furniture supplier. The specifications described the following business rules: **a customer who makes a monthly purchase that is higher than the average amount of monthly purchases for all customers** will be considered a **Preferred Customer**.

The Preferred Customer will be identified when making a purchase, and will be **immediately given a gift or major discount once a month**.

The defect introduced into the system (due to a poor understanding of the algorithm to set up for this requirement) involved **only taking into account the average amount of current purchases and not the customer's monthly history**. At the time of the software failure, the cash register was identifying far **too many Preferred Clients, resulting in a loss for the company**.

Development Life Cycle

Software life cycle process that contains the activities of **requirements analysis, design, coding, integration, testing, installation, and support for acceptance** of software products.

Depending on the **business model** of your organization, you will have to allow for **varying degrees of effort in identifying and correcting defects.**

Airbus, Boeing, Bombardier, and Embraer to have **identified and corrected all the defects** in the software for their airplanes **before we board them!**

Research Studies have come to the following conclusions:



- The scope of most defects is very **limited** and **easy to correct**.
- Many **defects occur outside of the coding activity** (e.g., requirement, architecture activities).
- **Poor understanding of the design** is a recurrent problem in programming error studies.
- It is a good idea to **measure** the number and origin of **defects** in your organization **to set targets for improvement**.

SQA need to develop a classification of the causes of software error by category.



- 1) problems with defining requirements.**
- 2) maintaining effective communication between client and developer.**
- 3) deviations from specifications.**
- 4) architecture and design errors.**
- 5) coding errors (including test code).**
- 6) non-compliance with current processes/procedures.**
- 7) inadequate reviews and tests.**
- 8) documentation errors.**

SOFTWARE QUALITY



- *Conformance to established software requirements; the capability of a software product to satisfy stated and implied needs when used under specified conditions.*
- *The degree to which a software product meets established requirements; however, quality depends upon the degree to which those established requirements accurately represent stakeholder needs, wants, and expectations*



SOFTWARE QUALITY ASSURANCE

Software Engineering

The systematic application of **scientific and technological knowledge, methods, and experience** for the design, implementation, testing, and documentation of software.

Quality Assurance

- 1) a planned and systematic pattern of all actions necessary to provide adequate confidence that an **item or product conforms to established technical requirements**;
- 2) a set of activities designed to evaluate the process by which products are developed or manufactured;
- 3) the planned and systematic activities implemented within the quality system, and demonstrated as needed, **to provide adequate confidence that an entity will fulfil requirements for quality.**

Software Quality Assurance



Definition

A set of activities that define and assess the adequacy of software processes to provide evidence that establishes confidence that the software processes are appropriate for and produce software products of suitable quality for their intended purposes.

A key attribute of SQA is the objectivity of the SQA function with respect to the project.

The SQA function may also be organizationally independent of the project; that is, free from technical, managerial, and financial pressures from the project.

SQA Elements



- The need to plan the **quality aspects of a product or service.**
- **Systematic activities** that tell us, **throughout the software life cycle**, that certain corrections are required.
- The quality system is a complete system that must, in the context of quality management, allow for the setting up of a **quality policy and continuous improvement;**
- QA techniques that **demonstrate the level of quality reached** so as to instil confidence in users and lastly.
- **Demonstrate that the quality requirements defined for the project**, for the change or by the software department have been met.

BUSINESS MODELS AND THE CHOICE OF SOFTWARE ENGINEERING PRACTICES



Business Model

A business model describes the rationale of **how an organization creates, delivers, and captures value** (economic, social, or other forms of value).

The essence of a business model is that it **defines the manner by which the business enterprise delivers value to customers, entices customers to pay for value, and converts those payments to profit.**

Choice of Software Practices



As expected, people from different business sectors chose software engineering practices that would lower the probability of their worst fears (quality and deadlines). Since their apprehensions are different, their practices are also different.

SUCCESS FACTORS

Foster Software Quality



- 1) **SQA techniques** adapted to the environment.
- 2) **Clear terminology** with regards to software problems.
- 3) An understanding and specific attention to each major category of software **error sources**.
- 4) An awareness of the SQA body of knowledge of the **SWEBOK as a guide for SQA**.

Factors that may Adversely Affect Software Quality



- 1) A lack of cohesion between SQA techniques and environmental factors in your organization.
- 2) Confusing terminology used to describe software problems.
- 3) A lack of understanding or interest for collecting information on software error sources.
- 4) Poor understanding of software quality fundamentals.
- 5) Ignorance or non-adherence with published SQA techniques.

Software quality assurance vs. software quality control



Quality control is defined as “a set of activities designed to **evaluate the quality of a developed or manufactured product**”

The main objective of **quality assurance** is to minimize the **cost of guaranteeing** quality by a variety of activities performed throughout the development and manufacturing processes/stages.

These activities prevent the causes of errors, and detect and correct them early in the development process.

Quality assurance activities substantially reduce the rate of products that do not qualify for shipment and, at the same time, reduce the costs of guaranteeing quality in most cases.

The objectives of SQA activities



Software development (process-oriented):

Software maintenance (product-oriented):

Software development (process-oriented):

- 1. Assuring an acceptable level of confidence that the software will **conform to functional technical requirements.**
- 2. Assuring an acceptable level of confidence that the software will **conform to managerial scheduling and budgetary requirements.**
- 3. **Initiating and managing of activities for the improvement and greater efficiency of software development and SQA activities.** This means improving the prospects that the functional and managerial requirements will be achieved while reducing the costs of carrying out the software development and SQA activities.

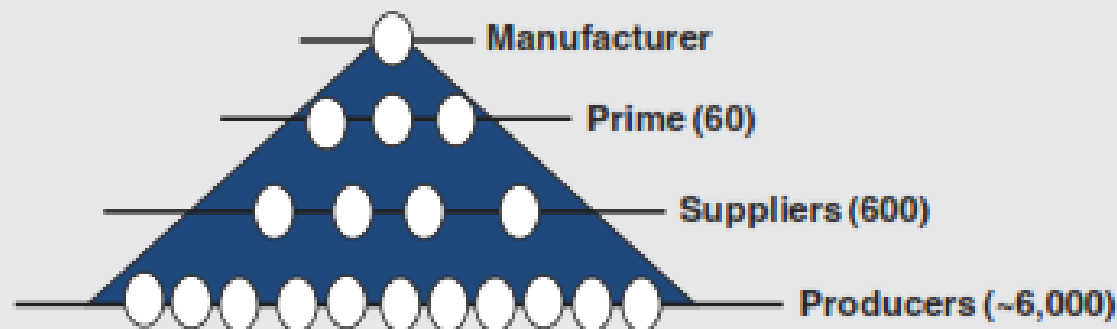
Software maintenance (product-oriented):

- 1. Assuring with an acceptable level of confidence that the software **maintenance activities will conform to the functional technical requirements.**
- 2. Assuring with an acceptable level of confidence that the software **maintenance activities will conform to managerial scheduling and budgetary requirements.**
- 3. **Initiating and managing activities to improve and increase the efficiency of software maintenance and SQA activities.** This involves improving the prospects of achieving functional and managerial requirements while reducing costs.

Quality Culture



A large Japanese electronic product manufacturing company uses a considerable number of suppliers. The supplier pyramid is made up of a first level with some sixty suppliers, a second level with a few hundred suppliers, and a third level with a few thousand very small suppliers.



A software defect for a component produced by a third-level supplier caused a loss of over \$200 million for this manufacturer.

Adapted from Shintani (2006) [SHI 06]

Setting up a **quality culture** and **software quality assurance (SQA) principles**, as stipulated in the standards, could help solve these problems.

Quality, influenced by organization's senior management and the organization's culture, has a cost, has a positive effect on profits, and must be governed by a code of ethics.

COST OF QUALITY



One of the major factors that explains **the** resistance to implementing quality assurance is the perception of its **high cost**.

As software engineers, responsible for informing administrators of the **risks that a company takes when not fully committed to the quality of its software**.

A practical manner of beginning the exercise is to **identify the costs of non-quality**.

It is easier to **identify potential savings by studying the problems caused by software**.

Costs of a project



- (1) Implementation costs
- (2) Prevention costs
- (3) Appraisal costs
- (4) The costs associated with failures or anomalies.

- If all development activities are error free, then 100% of costs could be implementation costs.
- Given that we make mistakes, we need to be able to identify them. The costs of detecting errors are appraisal costs (e.g., testing).
- Costs due to errors are anomaly costs.
- When we want to reduce the cost of anomalies, we invest in training, tools, and methodology. These are prevention costs.

The “cost of quality” is not calculated in the same way in all organizations.

- There is a certain amount of ambiguity between the notion of the cost of quality, the cost of non-quality, and the cost of obtaining quality.
- Most commonly used model at this time, costs of quality take into account the following five perspectives:
(1) prevention costs, (2) appraisal costs, (3–4) failure costs (internal during development, external on the client’s premises), and (5) costs associated with warranty claims and loss of reputation caused by non-quality.

calculation of the cost of quality in this model is as follows:



Quality costs = Prevention costs
+ Appraisal or evaluation costs
+ Internal and external failure costs
+ Warranty claims and loss of reputation costs

Prevention costs: This is defined as the cost incurred by an organization to prevent the occurrence of errors in the various steps of the development or maintenance process.

- For example, the cost of training employees, the cost of maintenance to ensure a stable manufacturing process, and the cost of making improvements.

Appraisal costs: The cost of verifying or evaluating a product or service during the different steps in the development process. Monitoring system costs (their maintenance and management costs).

Internal failure costs: The cost resulting from anomalies before the product or service has been delivered to the client. Loss of earnings due to non-compliance (cost of making changes, waste, additional human activities, and the use of extra products).

External failure costs: The cost incurred by the company when the client discovers defects. Cost of late deliveries, cost of managing disputes with the client, logistical costs for storing the replacement product or for delivery of the product to the client.

Warranty claims and loss of reputation costs: Cost of warranty execution and damage caused to a corporate image as well as the cost of losing clients.

The first objective of the SQA is to convince management that there are proven benefits to SQA activities.

“identifying an error early in the process can save a lot of time, money and effort.”



Figure 2.2 Costs of propagating an error¹ [JON 00].

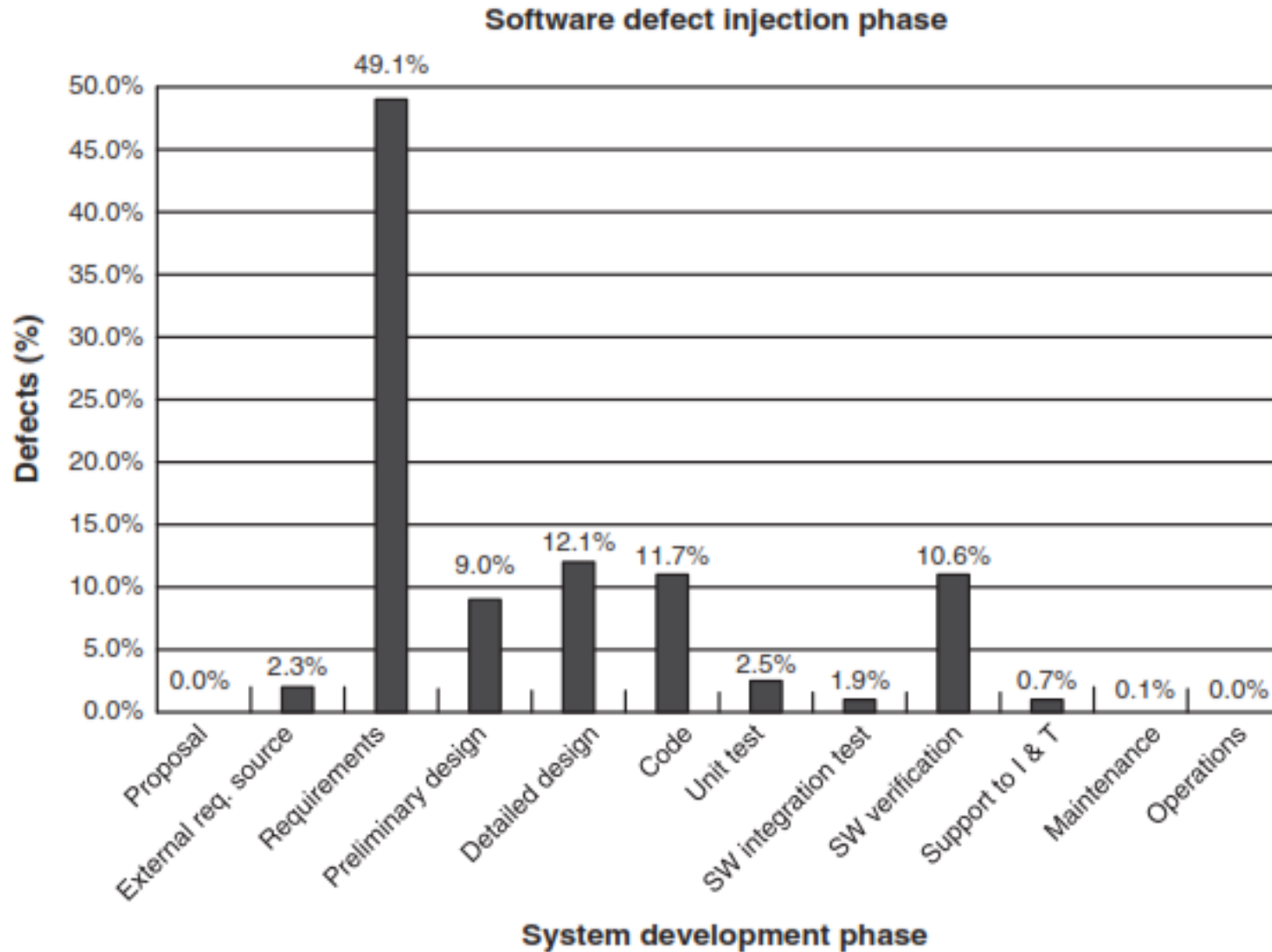


Figure 2.3 Defect injection distribution during the life cycle [SEL 07].