Tribhuvan University

Institute of Engineering

Pulchowk Campus

Department of Electronics and Computer Engineering

Software Engineering

Chapter Seven,Eight,Nine,Ten

by

Er. Bishwas Pokharel

Lecturer, Kathford Int'l College of Engg. and Mgmt.

Date: 14th Feb, 2018

Verification:

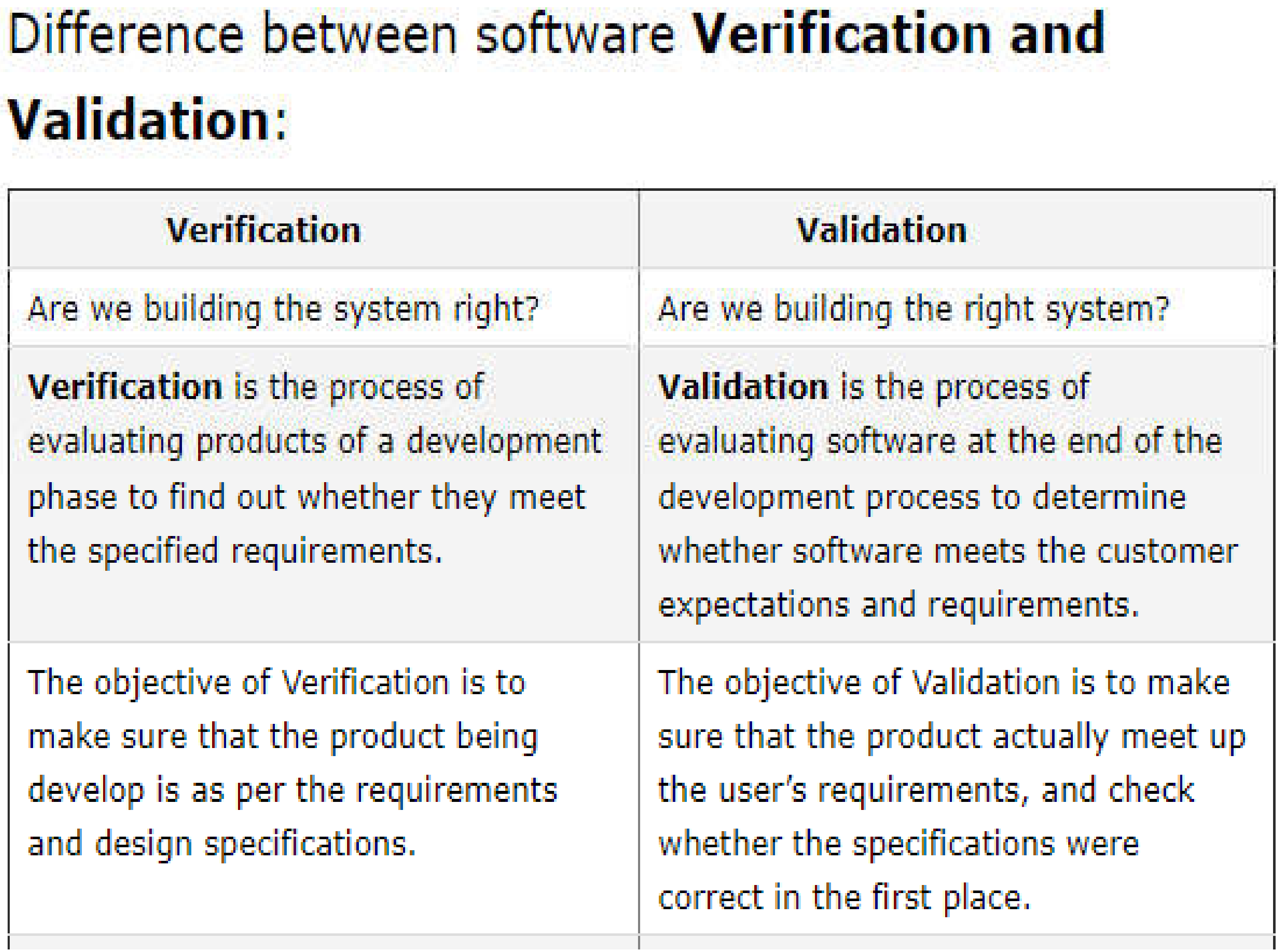
"Are we building the product right”.

- The software should conform to its specification.
- To ensure that it meets it specified functional & non functional requirements.

- **Validation:**

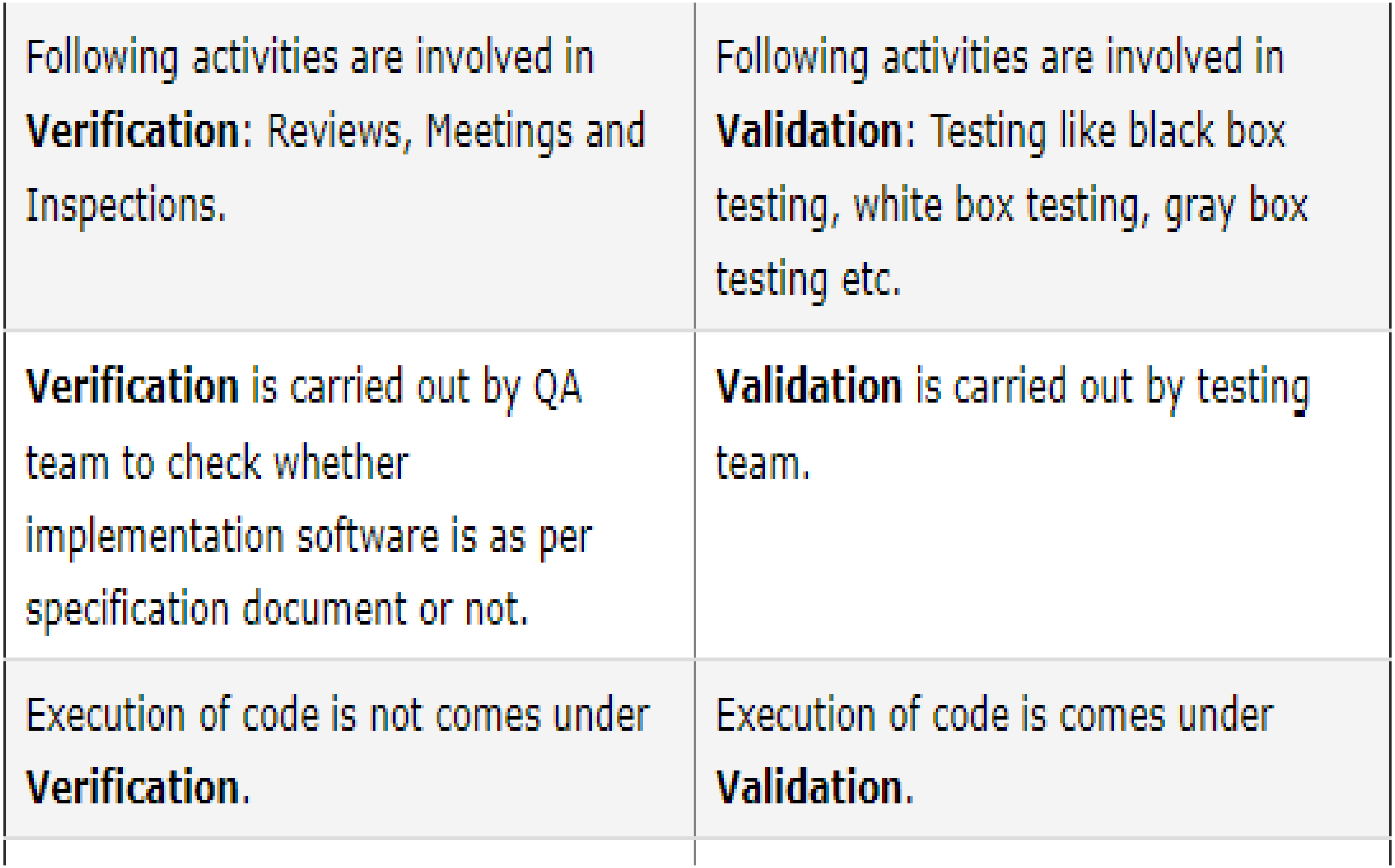
"Are we building the right product”.

- It is the more general process.
- The software should do what the user really requires.

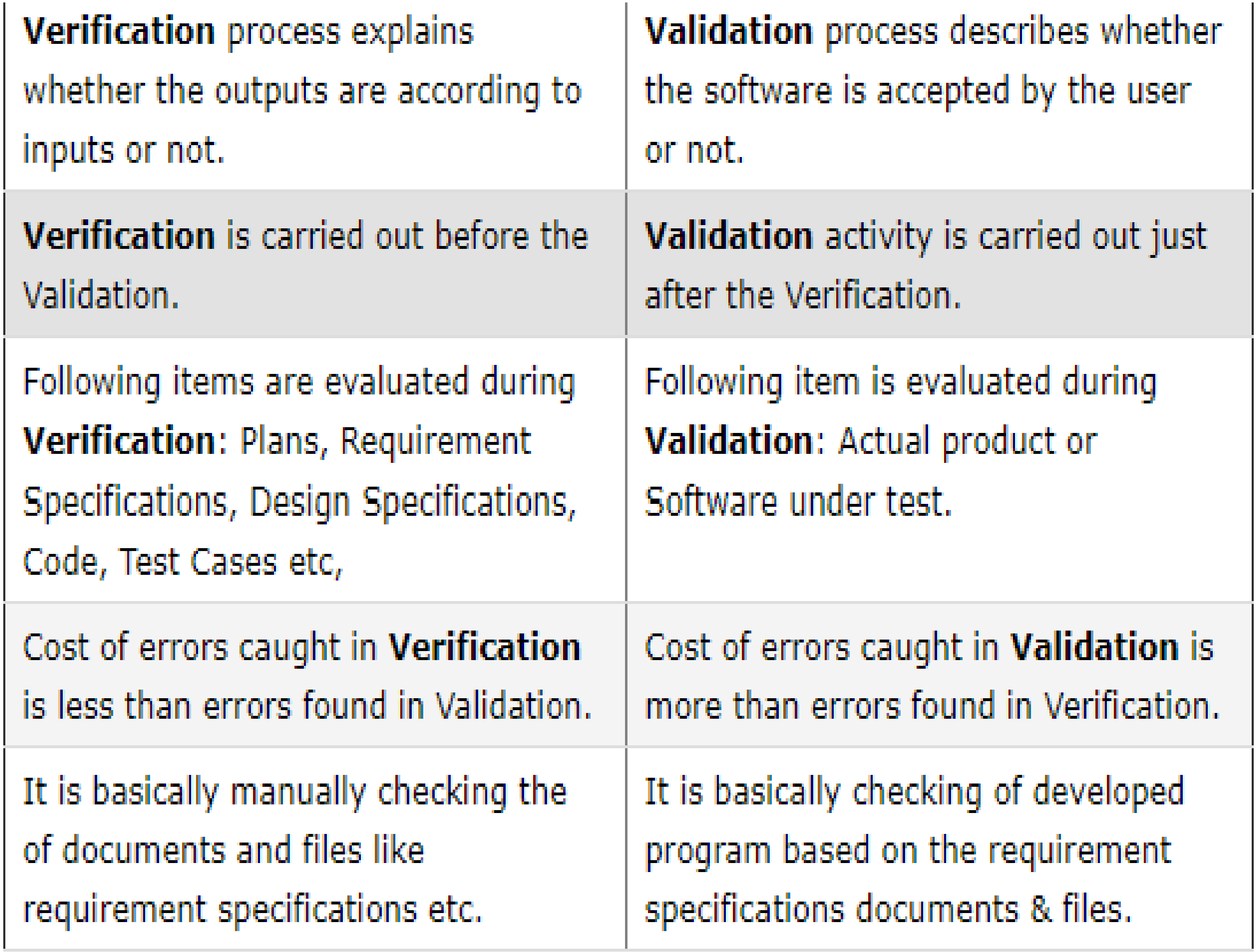


Difference between software **Verification and Validation**:

Verification	Validation
Are we building the system right?	Are we building the right system?
Verification is the process of evaluating products of a development phase to find out whether they meet the specified requirements.	**Validation** is the process of evaluating software at the end of the development process to determine whether software meets the customer expectations and requirements.
The objective of Verification is to make sure that the product being develop is as per the requirements and design specifications.	The objective of Validation is to make sure that the product actually meet up the user's requirements, and check whether the specifications were correct in the first place.



Following activities are involved in **Verification**: Reviews, Meetings and Inspections.	Following activities are involved in **Validation**: Testing like black box testing, white box testing, gray box testing etc.
Verification is carried out by QA team to check whether implementation software is as per specification document or not.	**Validation** is carried out by testing team.
Execution of code is not comes under **Verification**.	Execution of code is comes under **Validation**.



Verification process explains whether the outputs are according to inputs or not.	**Validation** process describes whether the software is accepted by the user or not.
Verification is carried out before the Validation.	**Validation** activity is carried out just after the Verification.
Following items are evaluated during **Verification**: Plans, Requirement Specifications, Design Specifications, Code, Test Cases etc,	Following item is evaluated during **Validation**: Actual product or Software under test.
Cost of errors caught in **Verification** is less than errors found in Validation.	Cost of errors caught in **Validation** is more than errors found in Verification.
It is basically manually checking the of documents and files like requirement specifications etc.	It is basically checking of developed program based on the requirement specifications documents & files.

Software Inspections

An **inspection** is ‘*a visual examination of a software product to detect and identify software anomalies, including errors and deviations from standards and specifications*’

What are software inspections (reviews)?

Meetings during which designs and code are reviewed by people other than the original developer.

Inspections

- A group of participants is nominated

- Participants must be familiar with inspections procedures so specific training is required to be able to participate.

- Each participant has a well defined role: moderator, reader, producer and recorder

Moderator -

- leads the inspection, schedules meetings, controls the meetings, reports inspection results, and follows up on rework issues.
- Moderators should be trained in how to conduct inspections, including how to keep participants with strong technical skills but low social skills from killing each other.

Note :A **work product** is a report, diagram, or collection of documents used by the business analyst during the requirements development process

Author -

- created or maintains the work product being inspected.

- The author may answer questions asked about the product during the inspection, and he also looks for defects.

- The author cannot serve as moderator, reader, or recorder

Recorder -

- Responsible for the synthesis of opinions expressed by all participants.

- Registers the following:
  a. physical location of defects found.
  b. description of defects category of each defect (from checklist)
  c. detects root cause (if identified)

Reader -

- describes the sections of the work product to the team as they proceed through the inspection.

- The reader may paraphrase what is happening in the product, such as describing what a section of code is supposed to do.

Inspector –

- attempts to find errors in the product.

- All participants actually are acting as inspectors, in addition to any other responsibilities.

- someone who is not involved in the project at all but who has the skill set and defect-detection abilities to be able to contribute usefully to inspecting any work product of this type.

Inspection checklists(programming inspection)

- Checklist of common errors should be used to drive the inspection.

- Error checklists are programming language dependent and reflect the characteristic errors that are likely to arise in the language.

- In general, the 'weaker' the type checking, the larger the checklist.

Inspection checks 1

Data faults	Are all program variables initialised before their values are used? Have all constants been named? Should the upper bound of arrays be equal to the size of the array or Size -1? If character strings are used, is a delimiter explicitly assigned? Is there any possibility of buffer overflow?
Control faults	For each conditional statement, is the condition correct? Is each loop certain to terminate? Are compound statements correctly bracketed? In case statements, are all possible cases accounted for? If a break is required after each case in case statements, has it been included?
Input/output faults	Are all input variables used? Are all output variables assigned a value before they are output? Can unexpected inputs cause corruption?

Inspection checks 2

Interface faults	Do all function and method calls have the correct number of parameters? Do formal and actual parameter types match? Are the parameters in the right order? If components access shared memory, do they have the same model of the shared memory structure?
Storage management faults	If a linked structure is modified, have all links been correctly reassigned? If dynamic storage is used, has space been allocated correctly? Is space explicitly de-allocated after it is no longer required?
Exception management faults	Have all possible error conditions been taken into account?

Inspector –

- attempts to find errors in the product.
- All participants actually are acting as inspectors, in addition to any other responsibilities.
- someone who is not involved in the project at all but who has the skill set and defect-detection abilities to be able to contribute usefully to inspecting any work product of this type.

Formal Technical Reviews

- Formal Technical review is a software quality assurance activity performed by software engineer.

Objectives of FTR

- FTR is useful to uncover error in logic, function and implementation for any representation of the software.

- The purpose of FTR is to ensure that software meets specified requirements.

- It is also ensure that software is represented according to predefined standards.
- It helps to review the uniformity in software development process.
- It makes the project more manageable.

Steps in FTR

1. The review meeting

Every review meeting should be conducted by considering the following constraints

- Involvement of people: Between 3 and 5 people should be involve in the review.
- Advance preparation: Advance preparation should occur but it should be very short that is at the most 2 hours of work for each person can be spent in this preparation
- Short duration :The short duration of the review meeting should be less than two hour.

2. Review reporting and record keeping

- During the FTR, the reviewer actively record all the issues that have been raised.
- At the end of meeting these all raised issues are consolidated and review issue list is prepared.
- Finally, formal technical review summary report is produced.

3. Review guidelines

Guidelines for the conducting of formal technical review must be established in advance. These guidelines must be distributed to all reviewers, agreed upon, and then followed.

For example, Guideline for review may include following things

- Concentrate on work product only. That means review the product not the producers.
- Set an agenda of a review and maintain it.
- When certain issues are raised then debate or arguments should be limited

- Find out problem areas, but don’t attempt to solve every problem noted.

- Take written notes (it is for record purpose)

- Limit the number of participants and insists upon advance preparation.

- Develop a checklist for each product that is likely to be reviewed.

- Allocate resources and time schedule for FTRs in order to maintain time schedule.

- Conduct meaningful trainings for all reviewers in order to make reviews effective.

- Reviews earlier reviews which serve as the base for the current review being conducted.

Capability Maturity Model(CMM)

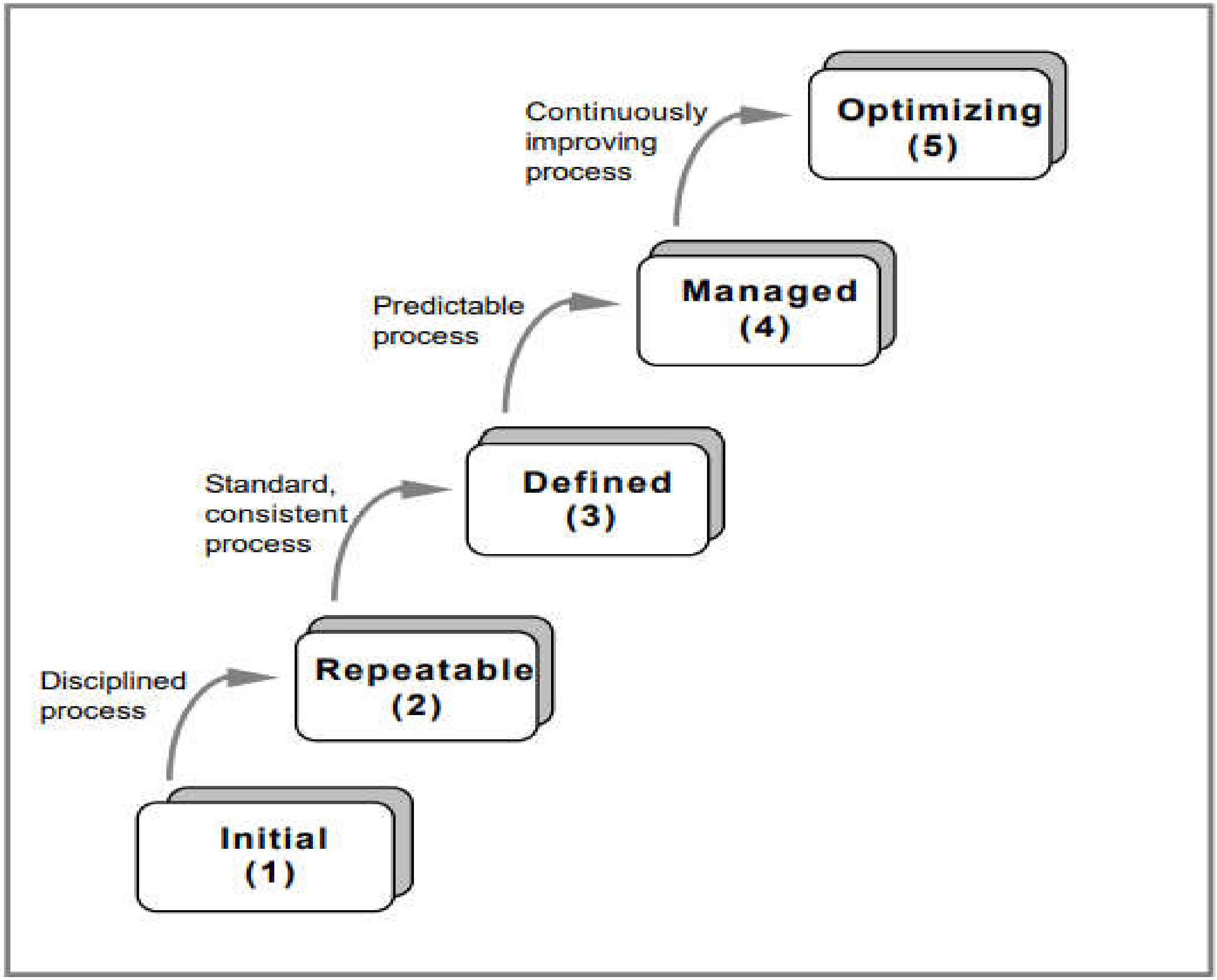
- A framework that describes the key elements of an effective software process.
- The CMM describes an evolutionary improvement path from an ad hoc, immature process to a mature, disciplined process.
- The CMM covers practices for planning, engineering, and managing software development and maintenance.
- When followed, these key practices improve the ability of organizations to meet goals for cost, schedule, functionality, and product quality.

- As organizations establish and improve the software processes by which they develop and maintain their software work products, they progress through levels of maturity.

- Each maturity level provides a layer in the foundation for continuous process improvement.

- Achieving each level of the maturity model institutionalizes a different aspect in the software process, resulting in an overall increase in the process capability of the organization

The Five Levels of Software Process Maturity



1) Initial The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.

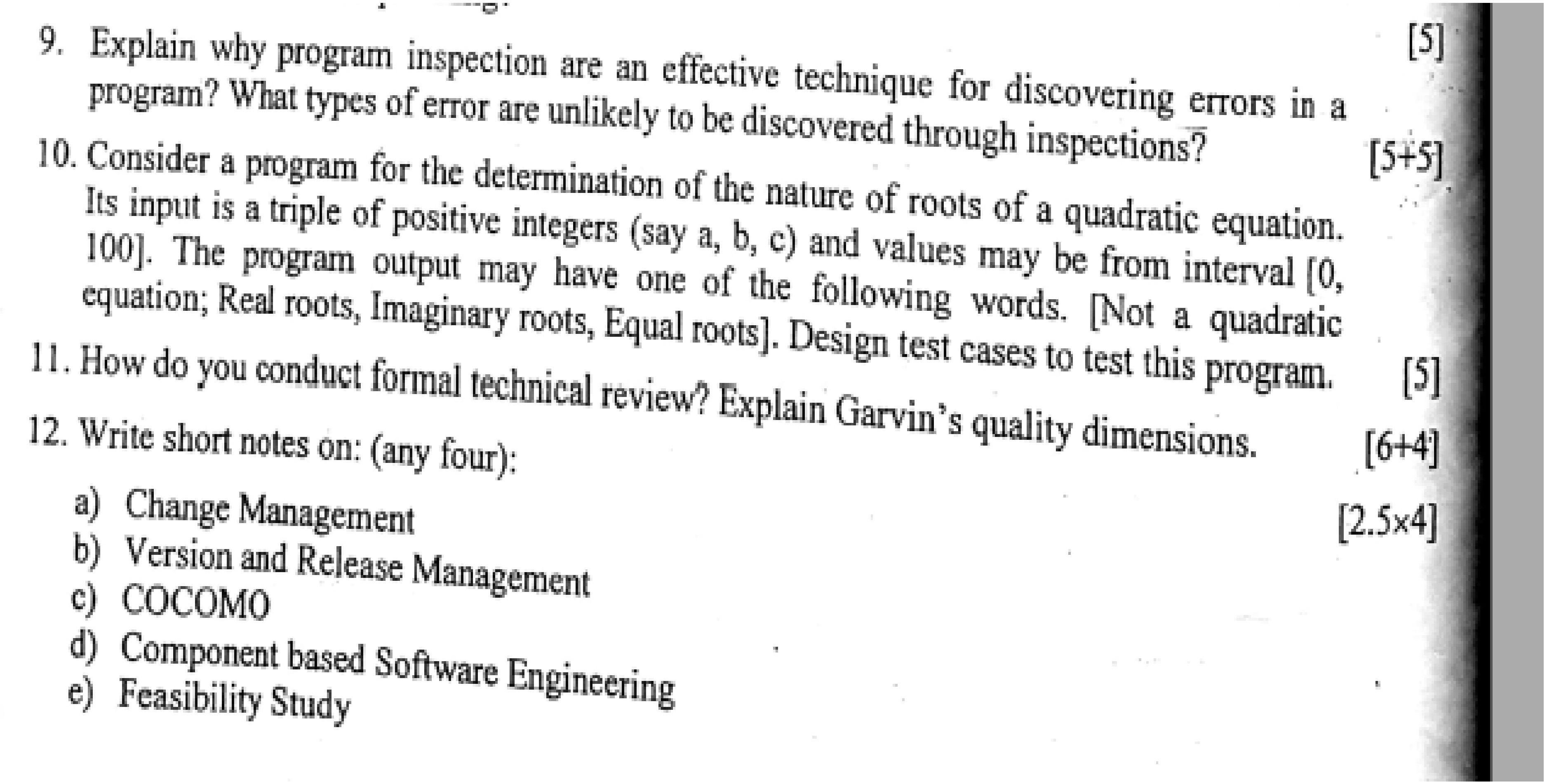
2) Repeatable Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.

3) Defined The software process for both management and engineering activities is documented, standardized, and integrated into a

standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.

4) Managed Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.

5) Optimizing Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies



9. Explain why program inspection are an effective technique for discovering errors in a program? What types of error are unlikely to be discovered through inspections? [5]

10. Consider a program for the determination of the nature of roots of a quadratic equation. Its input is a triple of positive integers (say a, b, c) and values may be from interval [0, 100]. The program output may have one of the following words. [Not a quadratic equation; Real roots, Imaginary roots, Equal roots]. Design test cases to test this program. [5+5]

11. How do you conduct formal technical review? Explain Garvin's quality dimensions. [5]

12. Write short notes on: (any four): [6+4]

a) Change Management [2.5×4]
b) Version and Release Management
c) COCOMO
d) Component based Software Engineering
e) Feasibility Study

Look pdf 1 for testing and pdf2 for testing numerical and software maintenance, software quality model and VVI _chapter.doc for vvi question for examination.

Assessment: falgun 11 Friday (All chapter)

if you cheat you will get 8 marks (no excuse!!)

else

19-20 marks

Thank You!!!