What is inspection?

- *Inspections are critical examinations of software artifacts by human inspectors.*
- *The purpose behind this inspection is discovering and fixing the faults in software.*
- *Inspection is non-testing way to eliminate the faults from the software.*

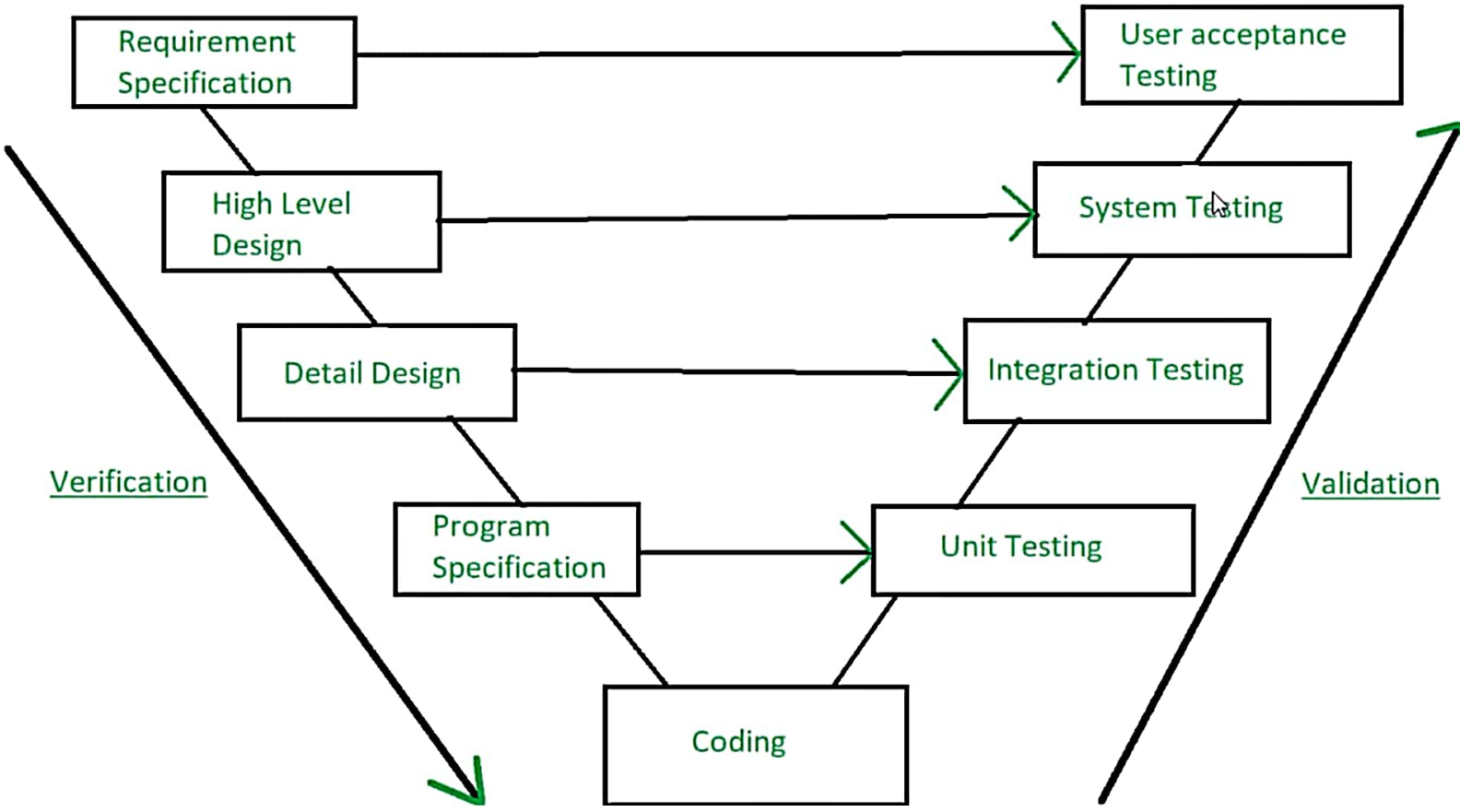
Different non-testing defect removal methods

- **Requirement inspections**
- **Design Inspection**
- **Code Inspection**
- **Test Case Reviews**
- **User Documentation Review**

Defect Seeding

- Will full insertion of defects in to the software artifacts just to check how to inspection process in going on how the inspection team is performing.

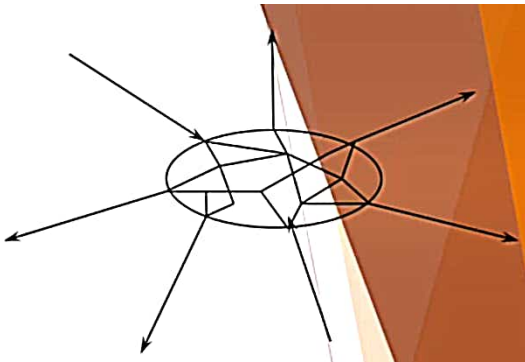
Verification	Validation
Verification means **Are we building the software right?**	Validation means **Are we building the right software ?**
Verification is the static testing.	Validation is the dynamic testing.
It does *not* include the execution of the code.	It includes the execution of the code.
Methods used in verification are reviews, walkthroughs, inspections and desk-checking.	Methods used in validation are Black Box Testing, White Box Testing and non-functional testing.
It checks whether the software conforms to specifications or not.	It checks whether the software meets the requirements and expectations of a customer or not.


Requirement Specification
User acceptance Testing
High Level Design
System Testing
Detail Design
Integration Testing
Verification
Validation
Program Specification
Unit Testing
Coding

System Testing

- **System Testing is a level of testing that validates the complete and fully integrated software product.**
- **The purpose of a system test is to evaluate the end-to-end system specifications.**
- **System Testing is a black-box testing.**
- **System testing categories based on: Who is Doing the Testing?**
- **System testing categories based on: Functional/Non-Functional Requirements?**

Component testing



- Lowest level System or a application
- Tested in isolation
- Most thorough look at detail
  - Error handling
  - Interfaces
- Usually done by programmer
- Also known as unit, module, program testing

What is Test case?

- Set of conditions, pre-conditions and steps under which we check If the software or some part of software is working or not.

Dummy Text Case	
Test Case ID:	1
TC Name	Login Account
Test Case Description	This test case tests the login account process
Dependency or Prerequisite	1. Connection is database properly established. 2. Account Is created or not
Steps	1. Application displays the link to login on start 2. User Press the link 3. New forum asking username and password appears 4. If correct information enter user can login 5. After login user can look his profile and other private data
Expected Result	Successfully login if the ID password is correct and above mentions requirements are fulfilled.
Actual Result	1. Logged in on correct information. 2. Logged in fail on wrong information.
Estimated time	Less than 1 second – On internet speed of 150kbps
Bugs, Errors	Nil

SPI

- Software process improvement is a set of activities that will lead to a better software process and in result higher-quality software delivered in a more timely manner.

- The people who champion SPI come from three groups: technical managers, software engineers, and individuals who have quality assurance responsibility.

SPI

- The approach to SPI is iterative and continuous, but it can be viewed in five QUICK LOOK steps:

- (1) assessment of the current software process,

- (2) education and training of practitioners and managers,
- (3) selection and justification of process elements, software engineering methods, and tools,
- (4) implementation of the SPI plan,
- (5) evaluation and tuning based on the results of the plan.

Assessment and Gap Analysis in SPI

- Assessment allows you to analyze the current working process, You find out the strengths and weaknesses of the current process in the assessment phase.

- The difference between local application and best practice represents a “gap” that offers opportunities for improvement.

Education in SPI

- when an SPI framework is introduced. It follows that a key element of any SPI strategy is education and training for practitioners, technical managers and more senior managers who have direct contact with the software organization. Three types of education and training should be conducted:

- **Generic concepts and methods.** Directed toward both managers and practitioners, this category stresses both process and practice.

Education in SPI

- **Specific technology and tools.** Directed primarily toward practitioners, this category stresses technologies and tools that have been adopted for local use. For example, if UML has been chosen for analysis and design modeling, a training curriculum for software engineering using UML would be established.

- **Business communication and quality-related topics.** Directed toward all stakeholders, this category focuses on "soft" topics that help enable better communication among stakeholders and foster a greater quality focus.

Selection & Justification in SPI

- First, you should choose the process model that best fits your organization, its stakeholders, and the software that you build.
- Secondly if you chose something you should have a proper justification for choosing with authentic arguments.

Installation/ Migration in SPI

- Installation is the 1st place where we see the impact of the SPI we have implemented.
- Sometimes changes do not work the way we want, so we have to change the whole SPI.
- Sometimes we don't have any need to change the whole SPI but minor changes are required such changes are often referred to as process migration.

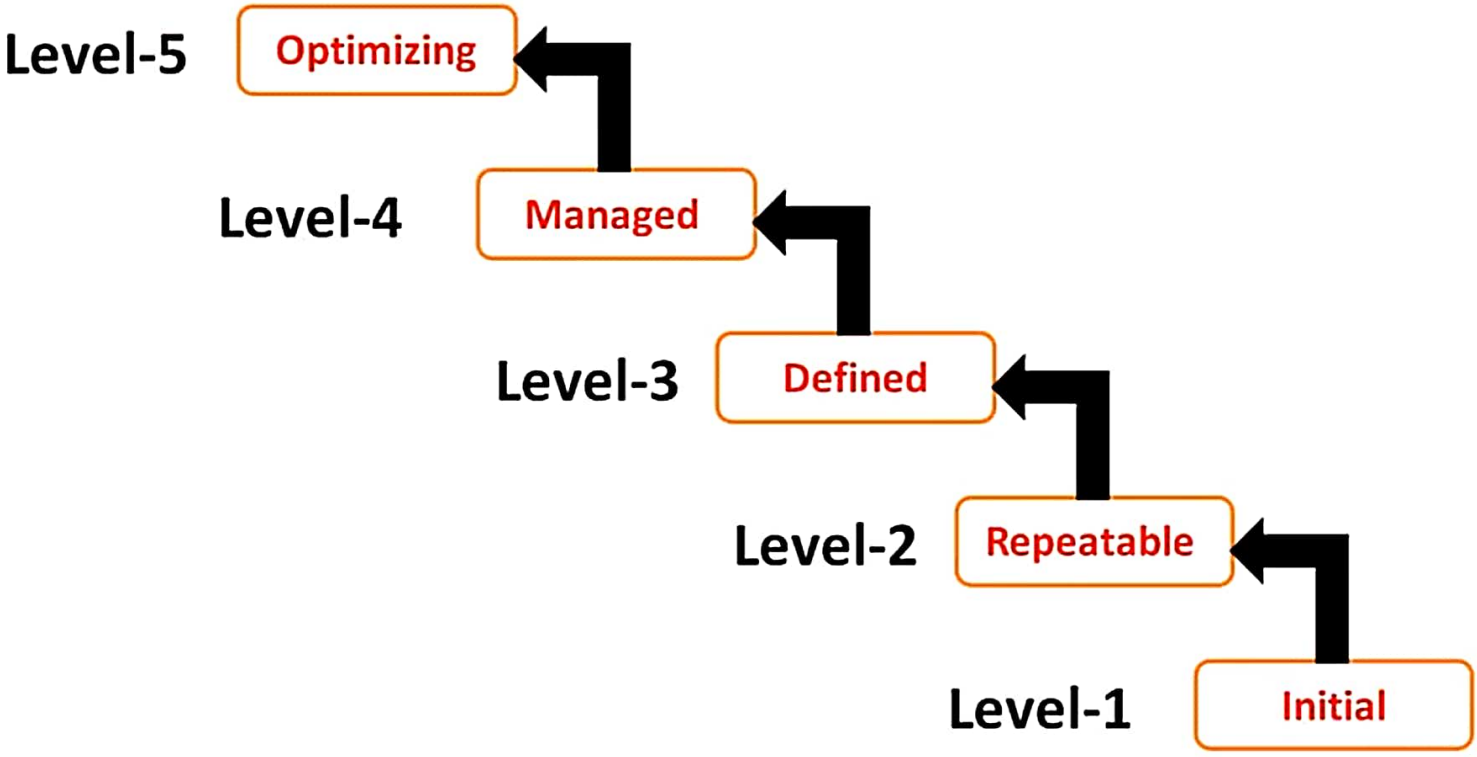
Evaluation in SPI

- At this stage we evaluate the changes, The impact of the changes the quality in software product.
- All in all at this stage we evaluate all the changes we got after implementing the SPI to any process.

1. **CMM** was developed and is promoted by the Software Engineering Institute (SEI), a research and development center.

2. **It is not a software process model. Capability Maturity Model** is used as a benchmark to **measure the maturity** of an organization's software process.

3. The model describes a **five-level** evolutionary path of increasingly organized and systematically more mature processes.


Level-5
Optimizing
Level-4
Managed
Level-3
Defined
Level-2
Repeatable
Level-1
Initial

Level One : Initial

Work is Performed Informally

A software development organization at this level is characterized by **ad hoc activities** (organization is not planned in advance).

Level Tow : Repeatable

Work is Planned and Tracked

This level of Software Development Organization has a basic and consistent project management processes to **track cost, schedule, and functionality**. The process is in place to **repeat** the earlier successes on projects with similar applications.

Level Three: Defined

Work is Well-Defined

At this level the software process for both management and engineering activities are **Defined** and **documented.**

Level Three: Defined

Work is Well-Defined

At this level the software process for both management and engineering activities are **Defined** and **documented.**

Level Four: Managed

Work is Quantitatively Controlled.

Software Quality Management: Management can effectively control the software development effort using precise measurements. At this level, organization set a quantitative quality goal for both software process and software maintenance.

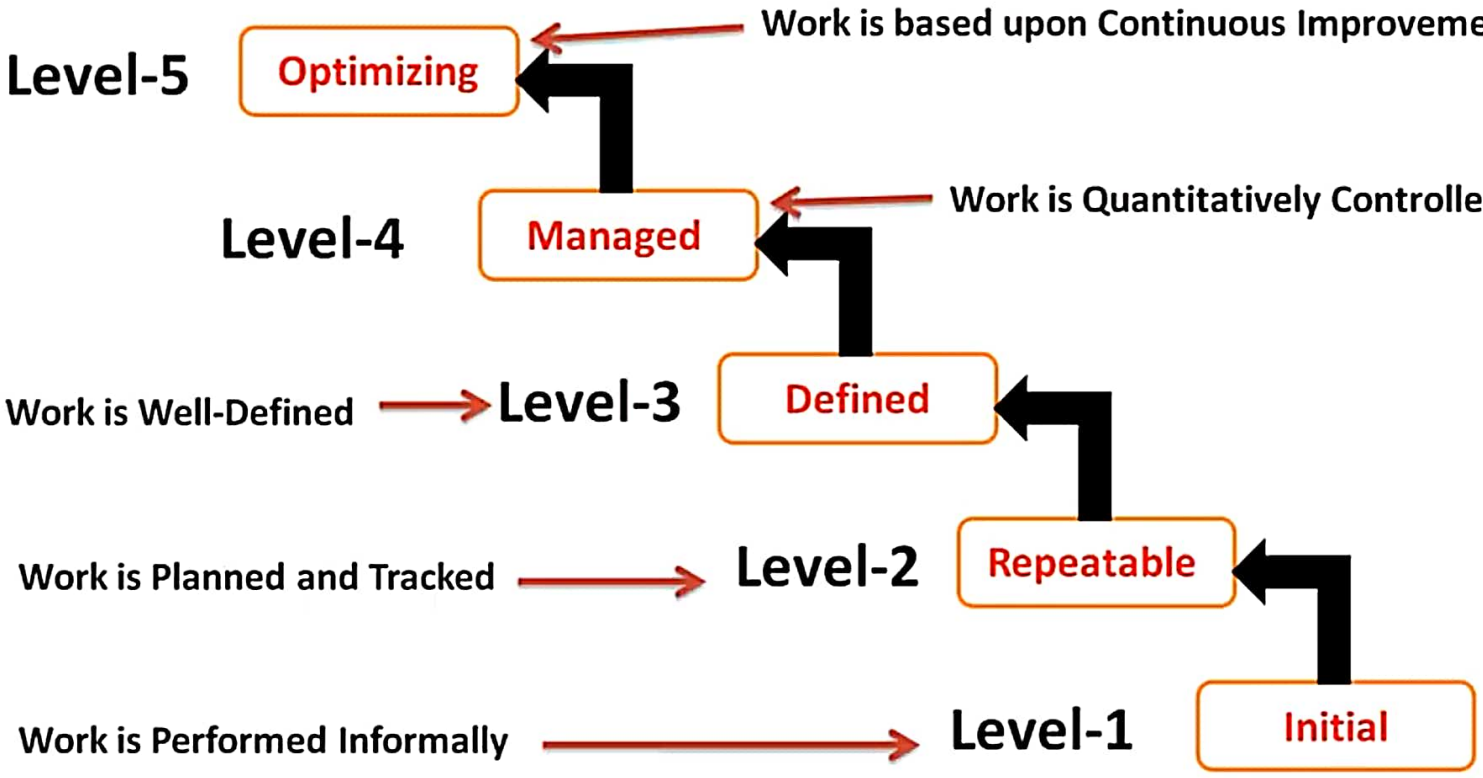
Quantitative Process Management: At this maturity level, the performance of processes is controlled using statistical and other quantitative techniques, and is quantitatively predictable.

Level Five: Optimizing

Work is based upon Continuous Improvement

The Key characteristic of this level is focusing on **continuously improving process** performance. Key features are:

- **Process Change Management**
- **Technology Change Management**
- **Defect Prevention**


Level-5
Optimizing
Work is based upon Continuous Improvement
Level-4
Managed
Work is Quantitatively Controlled.
Work is Well-Defined
Level-3
Defined
Work is Planned and Tracked
Level-2
Repeatable
Work is Performed Informally
Level-1
Initial